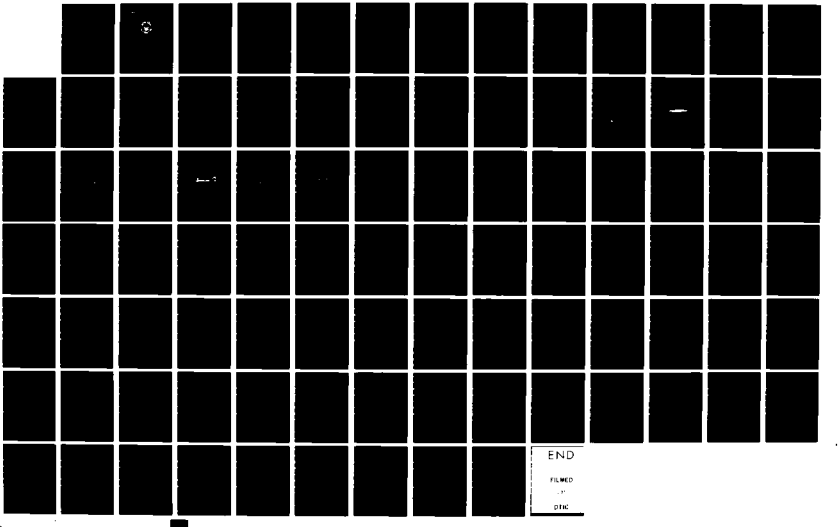
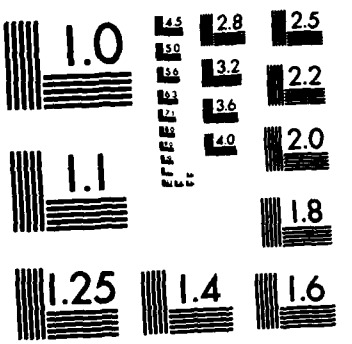
AD-A123 800 GRAPHIC SIMULATIONS OF THE POISSON PROCESS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA R J DAVISON OCT 82 1/1

UNCLASSIFIED F/G 9/2 NL





NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

GRAPHIC SIMULATIONS OF THE POISSON PROCESS

by

Richard John Davison

October 1982

Thesis Advisor: J. D. Esary

Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A123800	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Graphic Simulations of the Poisson Process		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis October 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard John Davison		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE October 1982
		13. NUMBER OF PAGES 90
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Simulation, Poisson process, Microcomputer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This paper presents the results of the development of graphic simulations of stochastic models using a microcomputer. Included are four computer programs simulating situations which are based on an underlying Poisson process. The programs are written in BASIC programming language for an Apple II Plus computer with a minimum of 48K of memory.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

Graphic Simulations of the Poisson Process

by

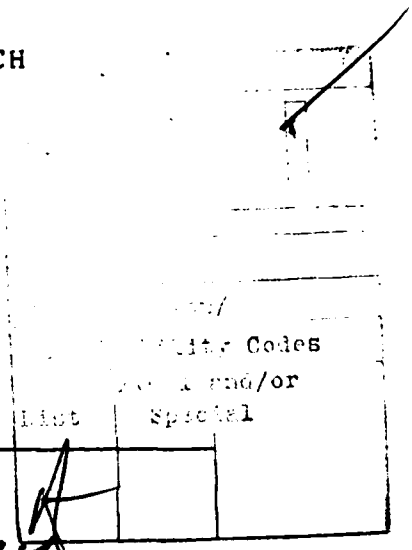
Richard John Davison
Lieutenant Commander, United States Coast Guard
B.S., University of Washington, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
OCTOBER 1982



Author: Richard J Davison

Approved by: James D. Esary
Thesis Advisor

Co-Advisor

Chairman, Department of Operations Research

W. M. Woods
Dean of Information and Policy Sciences

ABSTRACT

This paper presents the results of the development of graphic simulations of stochastic models using a microcomputer. Included are four computer programs simulating situations which are based on an underlying Poisson process. The programs are written in BASIC programming language for an Apple II Plus computer with a minimum of 48K of memory.

TABLE OF CONTENTS

I. INTRODUCTION..... 6

II. DISCUSSION..... 7

A. THE COMPUTER..... 7

B. THE LANGUAGE..... 7

C. SOFTWARE AIDS..... 8

D. PROGRAMMING METHODS..... 8

E. GRAPHIC SIMULATIONS..... 10

F. RANDOM NUMBERS AND TRANSFORMATIONS..... 11

G. THE RANDOM NUMBER GENERATOR..... 15

III. POISSON PROCESSES USER GUIDE..... 17

A. TABLE OF CONTENTS..... 18

B. LIST OF FIGURES..... 19

C. LIST OF TABLES..... 20

APPENDIX A. PROGRAM LISTINGS..... 36

A. POISSON PROCESS PROGRAM..... 37

B. MM1Q PROGRAM..... 49

C. DAM MODEL PROGRAM..... 61

D. DAMAGE MODEL PROGRAM..... 70

E. FISHPOND LOGO PROGRAM..... 75

F. MENU PROGRAM..... 82

G. SAMPLE HELP PROGRAM..... 84

H. STARTUP PROGRAM..... 87

BIBLIOGRAPHY..... 89

INITIAL DISTRIBUITON LIST..... 90

ACKNOWLEDGEMENTS

The author would like to thank professor James D. Esary and Major Jeffery L. Ellis for their continual encouragement and support throughout this project. A special thanks to the author's wife, Gerri, whose patience and understanding contributed considerably to a successful two years of study.

I. INTRODUCTION

This paper presents the results of an effort to develop visual simulations of a variety of stochastic models via a relatively new media, the microcomputer.

The intent of the effort was to create a real time perception of the models in action; to give the viewer an experience that could not otherwise be attained through text and classroom study. The microcomputer is suitable for this project due to its graphic capability, portability and accessability.

The theme for this project is the Poisson process. Four computer programs are provided that simulate the process either by itself or in applications. Represented are a simple and a superimposed Poisson process, an M/M/1 queue, a dam model and a damage model.

Descriptions of the programs and instructions to operate them are found in the User Guide. The remainder of this paper will touch upon several aspects of the project and programs that may be of interest to the general reader or of importance to those who may wish to pursue work of this nature. Program listings are found in Appendix A.

II. DISCUSSION

A. THE COMPUTER

An Apple II Plus computer with 48K of memory was used for this project. An Apple was selected because of its relatively easy accessability to anyone who may wish to run the programs. In addition, it is expected that the programs of this package will run with little or no modification on future generations of Apple hardware. The programs will run on an Apple III in the emulation mode.

B. THE LANGUAGE

The programs in this package are written in Applesoft BASIC, a subset of Microsoft BASIC with floating point and special graphics features. The selection of this language over other languages available, (including PASCAL, and FORTRAN), was a matter of convenience. Apple is a 'BASIC-speaking' computer, and there is a great deal of commercial software support for developing programs, graphics and other screen displays.

Although BASIC was used, Pascal was the first choice for this project. Its structured nature provides high legibility, aiding the interested reader as well as the original programmer in understanding the program. The author switched to BASIC after discovering that many specialized

graphics features required for the programs were not available in commercial Pascal graphics packages. Rather than make a compromise on the graphic presentation of the simulations, the author opted for an alternate language.

The type of programming language used is not critical. Speed is not essential, nor are there any special capabilities needed other than those provided by Applesoft BASIC. Most of the programming involves use of the computers' resident graphics package and a second special graphics package, to be described next.

C. SOFTWARE AIDS

Two commercial software packages were used to aid in producing the programs of this project; Printer II by Computer Systems Design, required to format numerical output in two of the programs, and the 'Hi-Res Character Generator' (HRCG) found in the Applesoft Tool Kit produced by Apple Computer, Inc. The Printer II package adds a final touch to the programs and is not absolutely necessary to successful production. The HRCG was essential. Without it, many of the graphics features would have been, at the least, very difficult to produce.

D. PROGRAMMING METHODS

Program listings are found in Appendix A. As will be noted, most of the programming structure deals with

preparation and execution of graphics. All of the programs are structured in a similiar fashion.

A typical program may be divided into several subroutines. The initialization portion of the program starts at line number 10000. Other important parts include; a main timing or clock routine starting at line 1000, plotting routines starting at 2000, a keyboard command routine starting at 3000, and in two cases a parameter change routine starting at line 4000.

The initialization portion of the program defines character strings, sets constants, and determines time to the first event of the program. The timing routine is similiar to a next-event structured simulation. The program loops within this routine, keeping count of each pass; the count representing time. In each pass a check is also made to see whether it is time for an event to occur. If so the program jumps to the plotting routine.

The plotting routine handles all the particulars needed to output on to the screen the effect of an event occurring, for example, a customer arriving at a queue, or an automobile passing by a particular point along a road. After the event occurs and while still in the plotting routine, a random number is obtained which is subsequently converted to represent time to the next event. Then, the program jumps back to the timing routine.

The keyboard routine is simply a 'control center' to filter commands intercepted from the keyboard. Keyboard commands not recognized are ignored. The parameter change routine starting on line 4000, is provided to allow the user to change mean interarrival times in two of the programs.

E. GRAPHIC SIMULATIONS

There are 53,760 locations arranged in a 280 by 192 array on the monitor screen available for program output. This presents a problem in truly representing a continuous random variable such as exponential interarrival times of the Poisson process. Fortunately, for the purposes of this project, this limitation is minor. The interarrival times as represented in the programs appear to belong to a Poisson process and the appearance is considered sufficient.

All of the programs in one way or another describe events and the interarrival times between events. The interarrival times shown or perceived are in fact geometric; the discrete analog of the exponential. The random number representing time to the next event is rounded up to the next highest integer to do this. The average interarrival time is slightly higher than the average that would result if the random numbers were truly exponential. The visual effect on the screen from this alteration is minimal.

F. RANDOM NUMBERS AND TRANSFORMATIONS

The following paragraphs briefly explain how the programs use random numbers to create Poisson simulations. The source of random numbers is the BASIC function RND. RND supplies uniform (Ø,1) random numbers which are then transformed in various ways to suit program objectives. The sequence of random numbers used in the programs are not repeatable, though this option was available.

Each of the four programs present simulations based on an underlying Poisson process, and therefore, all use exponential random numbers. To obtain the exponentials, uniform (Ø,1) random numbers are transformed through the inverse probability method. The variable 'y' is set equal to the uniform random number, and the variable 'x' is assigned the resultant exponential random number. The constant 'L' is the rate parameter;

$$F(x) = y = 1 - \exp(-L * x)$$

$$1 - y = \exp(-L * x)$$

$$\ln(1 - y) = -L * x$$

$$x = -\ln(1 - y) / L$$

or;

$$x = -\ln(y) / L$$

The above transformation is, of course, valid; F(x) is the distribution function of a continuous random variable and the inverse distribution function exists, therefore, the probability density of F(x) is uniform on the interval (Ø,1).

The program labeled 'Poisson Process' presents simulations of both a simple and a superimposed Poisson process. For the simple process, exponential random numbers are obtained one at a time. Initially, one exponential random number is generated. As the program progresses, a time counter is incremented and eventually becomes equivalent to the number, whereupon, an event is simulated. The program then generates another exponential random number, the counter is reset and the procedure begins again.

In the superimposed mode, two Poisson processes are active, each with a different rate parameter, (call them Type I and Type II with respective parameters L1 and L2). Two random numbers are obtained initially, one for each type. As the program progresses, a time counter is incremented until the value of the counter becomes equal to the smaller of the two numbers. Then, a simulation of the type of event that the smaller number is associated with, occurs. After the simulation, this smaller number is replaced with a new random number. The next event, (Type I or Type II), will occur when the counter becomes equivalent to the smaller of the two numbers now on hand; the new one or the one that was not involved with the last simulation.

Should the values of the two random numbers ever be equal, a uniform (Ø,1) random number is obtained and used to determine which of the two types of events will occur first. If the number is less than a predetermined value 'p', one

type of event occur, and if not, the second type of event will occur. The value 'p' is generated by the following equation where 'L1' and 'L2' are the rates of the two Poisson processes;

$$p = L1 / (L1 + L2)$$

The value 'p' is the probability that a Type I event will occur before a Type II event. Let 'T' be the time to the next event and 'T1' and 'T2' be independent random variables corresponding to the respective times that the Type I and Type II events will occur; then, the time to the next event will be;

$$T = \begin{cases} T1 & \text{with probability } p \\ T2 & \text{with probability } 1 - p \end{cases}$$

There is another transformation that takes place when mean interarrival times are changed by the user through keyboard commands. In that transformation, the mean interarrival times displayed on the screen are not those used in the transformation equations; they are converted first through an equation relating the exponential distribution to the geometric distribution. Below, the variable 'GM' is assigned the value obtained from the keyboard input, (and displayed on the screen), and the variable 'EM' is the

parameter used in the transformation equation, (EM = mean interarrival time = 1 / L; L = rate of the process);

$$EM = - \ln(1 - 1 / GM)$$

Thus if 'GM' is made equal to 5, then 'EM' will be assigned the value 4.48142Ø12. This procedure is included in the program only to synchronize the results presented in the statistics display with the time plot.

The M/M/1 queue model is built very much like the program 'Poisson Process' in that two exponential random numbers are needed, one for interarrival times, and one for service times. The numbers are obtained in the same way and the exponential to geometric conversion is identical. There is no concern for the possibility of a simultaneous occurrence of an arrival event and a departure event. Thus, if the two random variables should become equal the time plot included in the program indicates a simultaneous arrival and departure. Character animations representing arrivals and departures allow for the arrival to always occur first.

The two remaining programs involve the compound Poisson process. In the dam model, the two random numbers are used whenever an event occurs and therefore, are both replaced at the same time. One represents the interarrival time to the next event (arrival of a cloud) and forms the Poisson process. The second is an independent and identically distributed random variable independent of the interarrival

times, (the amount of rain that falls from each cloud). This second random variable is also exponential. Random variables with other distributions could have been used but the exponential was suitable for the graphic presentations.

The second compound Poisson process is found in the damage model program. Here, as with the dam model, two random numbers are obtained at the same time; both are exponential. One is used to form the Poisson process and represents times to successive events that cause damage. The amount of damage incurred is represented by the second random variable. Again, a random variable with some different distribution could have been used for amount of damage.

In both compound Poisson process models the random variables are not converted from geometric to exponential or vice versa.

G. THE RANDOM NUMBER GENERATOR

As noted previously, the BASIC function RND was used to obtain uniform (0,1) random numbers for program operation. It is not known how closely a sequence of these numbers conforms to a uniform distribuiton. The nature of the programs does not require exactness and therefore precision was not of great concern. Nevertheless, out of interest, two standard tests for distribution conformity and independence were run to obtain some measure of the generators' sufficiency.

The two tests run were the frequency test and the serial test. Each was run 100 times and for each of the 100 iterations, 1500 random numbers were used. The results were favorable. Comparison of the observed sequences with the expected characteristics of the uniform (0,1) distribution, showed a 3% rejection for the frequency test and a 5% rejection for the serial test. Using a chi-square goodness-of-fit test for each, the hypothesis that the distribution from which random numbers are obtained for the simulations is uniform (0,1), cannot be rejected at the 95% level of confidence.

For work requiring precision, further tests of RND would, of course, have to be conducted, and no definite conclusions concerning its adequacy can be drawn here.

III. POISSON PROCESSES USER GUIDE

Author

Richard J. Davison

Advisors

J. D. Esary

J. L. Ellis

TABLE OF CONTENTS

A. INTRODUCTION..... 21

B. EQUIPMENT REQUIREMENTS..... 21

C. STARTING UP..... 22

D. PROGRAM DESCRIPTIONS..... 24

E. ADDITIONAL KEYBOARD COMMANDS..... 32

LIST OF FIGURES

FIGURE 1: FIRST DISPLAY of INTRODUCTORY PROGRAM..... 22

FIGURE 2: SECOND DISPLAY of INTRODUCTORY PROGRAM..... 23

FIGURE 3: POISSON PROCESS SCREEN DISPLAY..... 25

FIGURE 4: MM1Q SCREEN DISPLAY..... 27

FIGURE 5: DAM MODEL SCREEN DISPLAY..... 29

FIGURE 6: DAMAGE MODEL SCREEN DISPLAY #1..... 30

FIGURE 7: DAMAGE MODEL SCREEN DISPLAY #2..... 31

LIST OF TABLES

TABLE 1: DEFINITION OF KEYBOARD COMMANDS..... 34

TABLE 2: REFERENCE TABLE FOR KEYBOARD COMMANDS..... 35

A. INTRODUCTION

The package of programs collectively labeled 'POISSON PROCESSES,' is designed for use on the Apple II or Apple II Plus microcomputer. The programs themselves are designed to provide the viewer with a real time perception of Poisson processes in action. Each simulates a different situation;

- 'Poisson Process,' is a simple demonstration of the occurrence of events in a Poisson manner.
- 'MM1Q,' is the simulation of a simple single-server queue with exponential interarrival times and exponential service times (the M/M/1 queue).
- 'Dam Model,' simulates one of the classic applications of the compound Poisson process.
- 'Damage Model,' represents the compound Poisson process applied to a cumulative damage model.

The following paragraphs provide an explanation of equipment requirements, procedures to follow to properly operate the microcomputer, and a more detailed description of each program.

B. EQUIPMENT REQUIREMENTS

An Apple II Plus computer (or Apple II with Applesoft card or Language System), with at least 48k of memory, a disk drive and controller, and a monitor (or television) are required to run these programs. The minimum memory size is

needed to provide two 'pages' of memory for graphics. An Apple is required not only because the programs are written in Applesoft Basic but because the programs use a special high resolution character generator program made for the Apple by Apple Computer Inc., to produce special effects on the screen.

C. STARTING UP

To run the programs, place the disk into the disk drive, and turn on the computer and monitor. The introductory

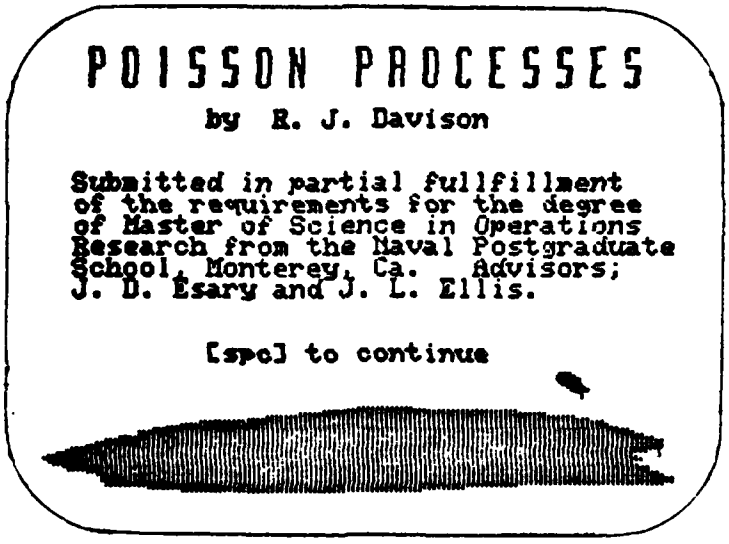


Figure 1: First Display of Indroductory Program

program, (FISHPOND LOGO), will be automatically loaded and run. It will take a few seconds before anything appears on the screen, as the startup program must first load the high

resolution character generator and character sets. The first presentation will look like Figure 1, above.

By pressing the space bar as directed in the first display the presentation will change to that shown in Figure 2;

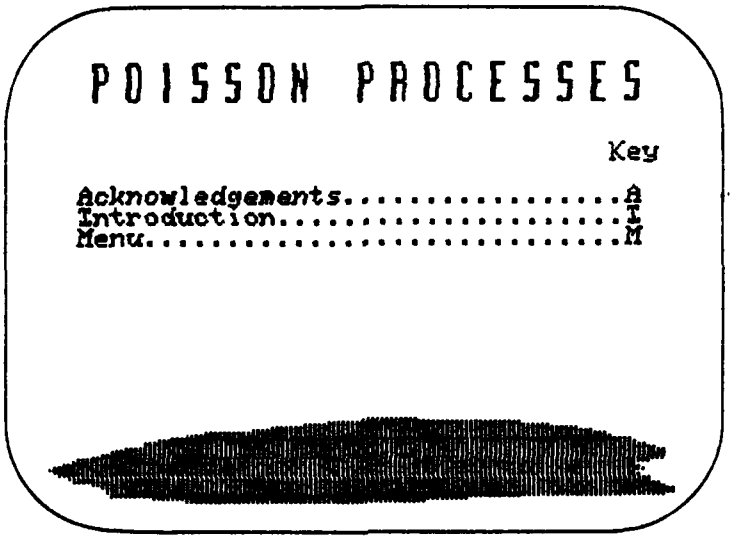


Figure 2: Second Display of Introductory Program

The 'Acknowledgements' section of the introductory program sights the use of commercial software used to run the programs. The 'Introduction' is a condensed version of the introduction in this guide and, therefore, will not be discussed here. Through the 'Menu,' the user can access each of the four programs which make up the package.

Once into a program, any of the other programs may be accessed either through the Menu or directly. To obtain the Menu at any time, simply press the 'M' key. To go to another

specific program directly, use the same key as called for in the Menu.

In addition, each program has its own 'Help' program which can be accessed by pressing the 'H' key. These Help programs give condensed versions of the following program descriptions as well as brief listings of keyboard commands which can be used in the respective programs.

D. PROGRAM DESCRIPTIONS

Each program is described separately on the subsequent pages. Each description includes a brief explanation of what is happening in the program and includes a summary of the commands that can be used during the program operation.

Program 1

POISSON PROCESS

This program illustrates a simple Poisson process in action. The graphical representation is composed of three parts; a time plot, animated characters which move across the screen at random intervals of time, and an optional statistics display. The advance of time is represented by a plot of 'tic' (or time) marks. To better associate real time with program time consider the unit of time as a 'second.' Different types of marks are plotted for one 'second', five 'seconds,' and sixty 'seconds.' The occurrence of an event is marked with a diamond shape and an animated character (bus or automobile), moves across the screen from left to right.

When the program is first run the viewer will be looking at a Poisson process which is in fact a composition of two

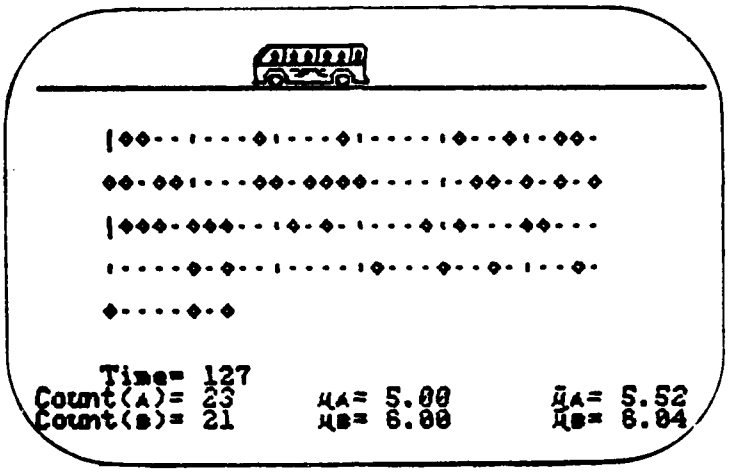


Figure 3: 'Poisson Process' Screen Display

Poisson processes; one process for the transit of automobiles and a second process for the transit of busses. This superimposed process may be decomposed by pressing the 'T' key. Subsequently, two lines of time marks will appear, the upper line for the auto and the lower line for the bus, (re-compose by pressing the 'U' key).

There are several other features to this program which the viewer may find useful. First, the user may alter the program so that only one type of event will occur; automobile or bus, (press the 'A' key for auto only; the 'B' key for bus only). Secondly, the user may change the mean interarrival times of either the auto, bus or both, (press the 'C' key and

follow the instructions). A statistics display is also provided, (press the 'D' key to turn on). The display shows a time count, and the theoretical and empirical mean interarrival times for the automobile and bus. The theoretical means are built into the program and may be changed as noted above. The empirical means change automatically with the passage of time, (the respective means are identified in the display by subscripts; 'A' stands for automobile and 'B' stands for bus; Press the 'O' key to turn the display off).

There are several ways to speed up the program so that, for example, one may watch the values of the empirical means converge to those of the theoretical means. First, the program has two built in speeds; slow and fast. The slow speed is the default, (press 'F' for the fast speed and 'S' to get back to the slow speed). Secondly, the program will run a little faster without the vehicles running across the screen, (press 'N'), and faster still if the time marks do not display, (press 'N' again; The statistics display must be on to turn the time marks off; Press the 'Y' key twice to return all screen representations).

As an example, a key-press sequence might proceed as follows; Press 'D' for the statistics display; Press 'N' to stop the vehicles from running across the screen; Press 'N' again to stop the 'tic' marks from being plotted. Now, only the statistics display will be showing, (Again, note that the

statistics display must be showing before the 'tic' marks can be deleted from the screen). To bring everything back, reverse the sequence; Press 'Y', then 'Y' again, then 'O' to delete the display.

Program 2

MM1Q

This program provides a visual display of the M/M/1 queue; a single-server queue with exponential interarrival times and exponential service times. As with the 'Poisson

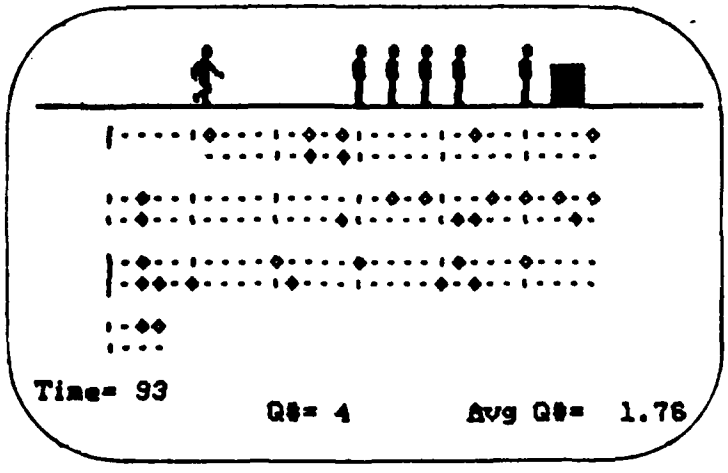


Figure 4: 'MM1Q' Screen Display

Process' model, this program too provides a three part graphical presentation; animated figures that move across the screen, 'tic' marks that record the passage of time and an optional statistics display.

The keyboard commands are nearly the same. The user may alter the mean interarrival time and service time, (press key 'C'), change the program speed, ('S' for slow and 'F' for fast), view a statistics display, ('D' to turn on and 'O' to turn off), and stop the character animation and time plot to speed up the program.

The screen presentation will show two lines of 'tic' marks. The upper line provides a representation of time and customer arrivals, which are marked by an open diamond shape. The lower line depicts service times and a departure event marked by a solid diamond shape. When there are no customers at the 'service counter' no service time marks will be plotted. It will be noted that the customer being serviced is close to the 'service counter' and not part of the queue.

The statistics display shows a time count, the queue size (Q#), and the average queue size (Avg Q#). The theoretical means are not shown. They may be determined however, by entering the parameter change routine, (press key 'C').

Program 3

Dam Model

This program portrays one of the classic applications of the compound Poisson process. In the model, clouds arrive in a random fashion and with each appearance a random quantity of rain falls. The cloud arrivals form a Poisson process, and the amounts of rain from the clouds are independent and

identically distributed random variables, independent of the cloud arrival times, (the rainfall amounts are exponential in this model). There is an outflow of water from the dam at a constant rate.

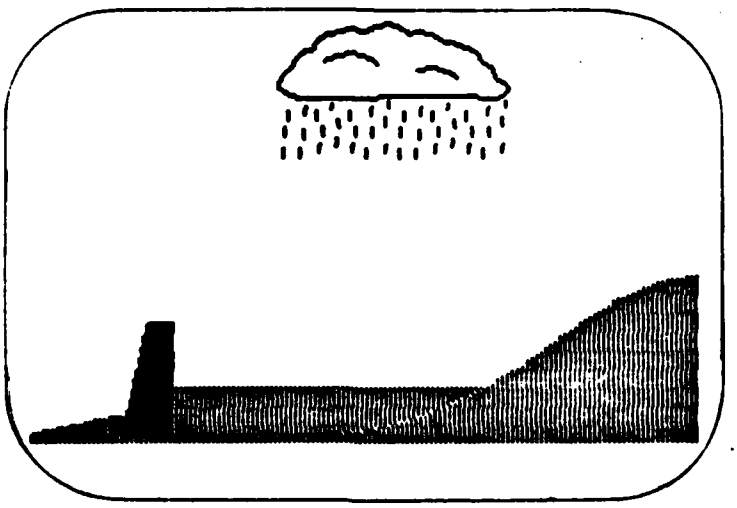


Figure 5: 'Dam Model' Screen Display

There are few keyboard commands for this program. As with the two previous models, there is a fast clock time and a slow clock time, ('F' and 'S'). In addition, the viewer may interrupt the program to raise or lower the water behind the dam, ('U' to raise the level, 'D' to lower the level, and 'N' to return to normal program operation). No statistics display is available, nor is there a parameter change routine.

Program 4

DAMAGE MODEL

This program depicts the compound Poisson process applied as a cumulative damage model. The graphical representation is composed of two parts; a time plot represented by horizontal line segments, and a balance beam or seesaw that provides a way to represent accumulation of damage.

At the beginning of the program, a horizontal line appears at the lower left hand corner of the border outline on the screen. The appearance of this line indicates the

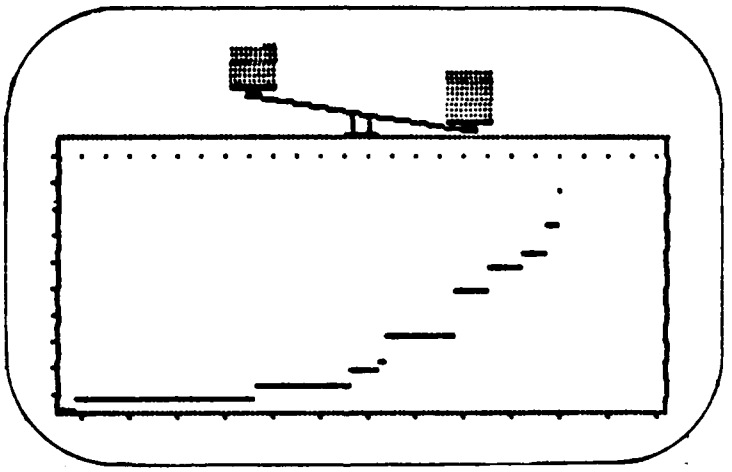


Figure 6: 'Damage Model' Screen Display #1

beginning of a lifecycle. The line will increase to a random length after which a new horizontal line will appear a random vertical distance above the old one. The new line will increase to a random length then another jump will occur, and

so on. The length of a line represents the interarrival time between events. The occurrence of a jump represents an event that causes damage, and the vertical magnitude of the jump represents the amount of damage incurred. The event arrivals form a Poisson process, and the amounts of damage are independent and identically distributed random variables, independent of the event arrival times, (the damage amounts are exponential in this model).

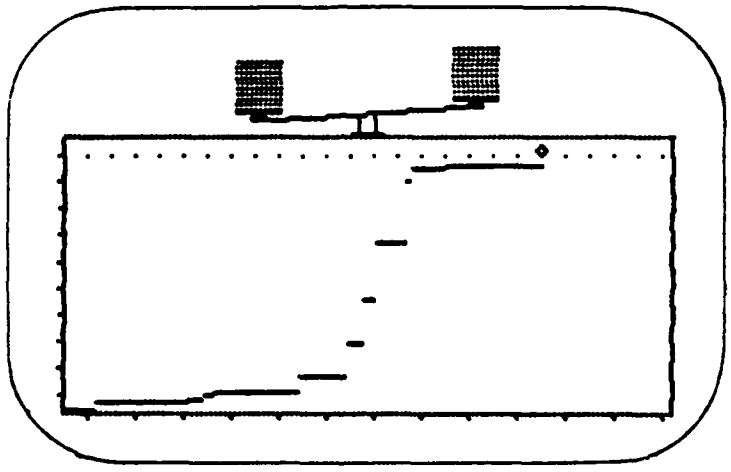


Figure 7: 'Damage Model' Screen Display #2

When a jump occurs, 'weight' is added to the left side of the seesaw. The amount of weight added corresponds to the amount of damage incurred. The dotted line near the top of the screen border represents a threshold; a point which when reached will signal the end of the lifecycle. The final jump to this level will add enough 'weight' to the seesaw to cause

it to swing from its initial position of rest to the opposite extreme; a failure has occurred. The program then resets the screen images and restarts with a new lifecycle.

Again, there are few keyboard commands. There is no fast/slow clock speed option or parameter change routine. Those options available are common to all programs and will be explained in the concluding paragraphs that follow.

E. ADDITIONAL KEYBOARD COMMANDS

In addition to the keyboard commands noted in the program descriptions, there are two that have not been mentioned which are common to all the programs. The first is the option to temporarily stop a program. This may be done by pressing the space bar. The program will do nothing further and the picture on the screen will remain motionless until the space bar is pressed again.

The second and final command is the option to quit the package of programs altogether. Pressing the 'Q' key will do it; the program will end and the screen will go blank. This action is effective in concluding a session with this package. Note however, that two further steps should be taken before the computer is used for further programming; 1) press the reset button to disable the high resolution character generator and clear memory of the character sets used in the programs, and 2) press the key sequence 'FP' to reset 'himem'. If these two steps are not carried out before

the computer is put to some other use the user may run out of memory sooner than expected. Of course an alternate and perhaps simpler approach to resetting the computer is to turn it off and then back on.

The two tables that follow provide of summary of keyboard commands. Table 1, gives the definitions of each command. Table 2, supplies commands available in each program.

Table 1: Definition of Keyboard Commands

Key	Definition
A	Run Poisson process for auto only
B	Run Poisson process for bus only
C	Change parameters
D	Turn statistics display on
D	Lower water level (Dam Model)
F	Run program fast
N	1st time; No character animation
	2nd time; no time plot
N	Normal program operation (Dam Model)
O	Turn statistics display off
Q	End session
S	Run program slow (default)
T	Run decomposed Poisson process (auto and bus)
U	Run superimposed Poisson process (auto and bus)
U	Raise water behind dam (Dam Model)
Y	Resume graphics (after 'N' is pressed one or more times)
spc bar	Stop program temporarily/resume program after temporary stop

Table 2: Reference Table for Keyboard Commands

Program	Keyboard Commands Available
Poisson Process	A,B,C,D,F,N,O,Q,S,T,U,Y,spc bar
MM1Q	C,D,F,N,O,S,Q,Y,spc bar
Dam Model	D,F,N,S,Q,U,spc bar
Damage Model	Q,spc bar

APPENDIX A

PROGRAM LISTINGS

The program listings are written in Apple II Applesoft BASIC. The programs were listed using XLISTER, a Beagle Brothers, Inc. program.

A. POISSON PROCESS PROGRAM

```
---------------------
10    GOSUB 10000
    : REM INITIALIZE
---------------------
999   REM MAIN ROUTINE; CLOCK
---------------------
1000  IF TM = 30 OR TM = 0 THEN POKE 37,( PEEK (37) + 1)
  * : IF TD = 1 THEN PRINT C$
---------------------
1010  IF TD > 1 AND PEEK (37) > 14 THEN POKE 37,15
  * : PRINT C$
  * : POKE 37,14
  * : GOTO 1030
---------------------
1020  IF PEEK (37) > 14 THEN POKE 37,14
---------------------
1030  IF TD = 3 THEN 1060
---------------------
1040  IF T1 = 0 THEN VT = 1
  * : GOSUB 1500
  * : GOTO 1120
  * : REM MOVE AUTO
---------------------
1050  IF TD = 2 THEN 1070
---------------------
1060  IF T2 = 0 THEN VT = 2
  * : GOSUB 1500
  * : GOTO 1120
  * : REM MOVE BUS
---------------------
1070  IF NN = 0 THEN 1110
---------------------
1080  IF TM / 60 = INT (TM / 60) THEN PRINT MIN$(L);
  * : GOTO 1110
---------------------
1090  IF TM / 5 = INT (TM / 5) THEN PRINT DASH$(L);
  * : GOTO 1110
---------------------
1100  PRINT DOT$(L);
---------------------
1110  FOR I = 1 TO R
    : NEXT
    : IF TD = 1 THEN POKE 36,( PEEK (36) + 1)
---------------------
1120  IF TD < > 3 THEN T1 = T1 - 1
---------------------
1130  IF TD < > 2 THEN T2 = T2 - 1
---------------------
1140  TM = TM + 1
```

```
    : IF TM = 60 THEN TM = 0
----------------------
1150  IF PEEK ( - 16384) > 127 THEN GOSUB 3000
  * : REM KEY PRESSED TO STOP CLOCK
----------------------
1160  TT = TT + 1
    : IF TT > 32765 THEN PRINT Y$;P$
  * : GOTO 10
----------------------
1170  IF C1 > 0 THEN M1 = TT / C1
  * : IF TD = 3 THEN M1 = 0
----------------------
1180  IF C2 > 0 THEN M2 = TT / C2
  * : IF TD = 2 THEN M2 = 0
----------------------
1190  IF SD = 1 THEN CH = PEEK (36)
  * : CV = PEEK (37)
  * : PRINT Y$;A$"2"
  * : VTAB 18
  * : HTAB 11
  * : PRINT TT
  * : VTAB 19
  * : HTAB 35
  * : PRINT FOR 5,2;#M1
  * : VTAB 20
  * : HTAB 35
  * : PRINT FOR 5,2;#M2;A$"1"
  * : GOSUB 2600
  * : IF TD = 1 THEN POKE 36,( PEEK (36) - 1)
----------------------
1200  GOTO 1000
----------------------
1499  REM EVENT SUBROUTINES
----------------------
1500  IF NN = 0 THEN 2000
----------------------
1510  IF TD = 2 OR TD = 4 THEN PRINT DIAM$(1);
  * : GOTO 2000
----------------------
1520  IF TD = 3 THEN PRINT DIAM$(8);
  * : GOTO 2000
----------------------
1530  K = 4
    : IF VT = 2 THEN K = 1
----------------------
1540  IF TM / 60 = INT (TM / 60) THEN PRINT DIAM$(K + 3);
  * : GOTO 2000
----------------------
1550  IF TM / 5 = INT (TM / 5) THEN PRINT DIAM$(K + 2);
  * : GOTO 2000
----------------------
```

```
1560  PRINT DIAM$(K + 1);
---------------------
1999  REM   VEHICLE RUN SUBROUTINE
---------------------
2000  CH = PEEK (36)
    : CV = PEEK (37)
    : PRINT Y$
    : REM SAVE CURSOR POSITION
---------------------
2010  IF TD < > 3 AND VT = 1 THEN C1 = C1 + 1
  * : M1 = TT / C1
---------------------
2020  IF TD < > 2 AND VT = 2 THEN C2 = C2 + 1
  * : M2 = TT / C2
---------------------
2030  IF SD = 1 THEN PRINT A$"2"
  * : VTAB 19
  * : HTAB 11
  * : PRINT C1
  * : VTAB 19
  * : HTAB 35
  * : PRINT FOR 5,2;#M1
  * : VTAB 20
  * : HTAB 11
  * : PRINT C2
  * : VTAB 20
  * : HTAB 35
  * : PRINT FOR 5,2;#M2;A$"1"
---------------------
2040  IF N = 0 THEN 2500
---------------------
2050  FOR I = 1 TO 33
    : VTAB 3
    : HTAB I
    : PRINT CAR$(VT);
    : NEXT
    : VTAB 3
    : HTAB 33
    : PRINT E$
    : VTAB 4
    : HTAB 33
    : PRINT E$
---------------------
2499  REM NEXT EVENT SUBROUTINE
---------------------
2500  IF VT = 2 THEN 2540
---------------------
2510  T = - E1 * LOG ( RND (1))
---------------------
2520  T1 = INT (T) + 1
    : REM AUTO ARRIVAL TIME
```

```
----------
2530  IF VT = 1 THEN 2560
----------
2540  T = - E2 * LOG ( RND (1))
----------
2550  T2 = INT (T) + 1
    : REM BUS ARRIVAL TIME
----------
2560  IF T1 < > T2 THEN 2600
----------
2570  T = RND (1)
----------
2580  IF T < P THEN T2 = T2 + 1
----------
2590  IF T > P THEN T1 = T1 + 1
----------
2600  VTAB 6
    : HTAB 5
    : PRINT V$;
    : VTAB 16
    : HTAB 30
    : PRINT W$;
    : POKE 36,CH
    : POKE 37,CV
    : IF TD = 1 THEN POKE 36,( PEEK (36) + 1)
----------
2610  RETURN
----------
2999  REM KEYPRESS SUBROUTINE
----------
3000  CH = PEEK (36)
    : CV = PEEK (37)
----------
3010  IF PEEK ( - 16384) = 211 THEN R = 1000
  * : GOTO 3250
  * : REM RUN CLOCK SLOW
----------
3020  IF PEEK ( - 16384) = 198 THEN R = 1
  * : GOTO 3250
  * : REM RUN CLOCK FAST
----------
3030  IF PEEK ( - 16384) = 206 AND N = 0 AND SD = 1 THEN NN =
        0
  * : PRINT P$
  * : GOTO 3250
  * : REM SHOW STATS DISPLAY ONLY
----------
3040  IF PEEK ( - 16384) = 206 THEN N = 0
  * : GOTO 3250
  * : REM SHOW ALL BUT CHARACTER ANIMATION
----------
```

```
3050  IF PEEK ( - 16384) = 217 AND NN = 0 THEN NN = 1
  * : CH = 0
  * : CV = 5
  * : TM = 0
  * : IF TD = 1 THEN CV = 2
  * : GOTO 3250 REM IN DISPLAY ONLY MODE; BRING BACK TIMING
      PLOT
----------
3060  IF PEEK ( - 16384) = 217 THEN N = 1
  * : GOTO 3250
  * : REM NOT SHOWING CHARACTER ANIMATION;BRING IT BACK
----------
3070  IF PEEK ( - 16384) = 196 THEN SD = 1
  * : GOSUB 3500
  * : GOSUB 2600
  * : GOTO 3250
  * : REM SET DISPLAY MODE
----------
3080  IF PEEK ( - 16384) = 207 AND NN = 1 THEN SD = 0
  * : PRINT Y$
  * : VTAB 18
  * : HTAB 1
  * : PRINT F$
  * : GOSUB 2600
  * : GOTO 3250
  * : REM TURN OFF DISPLAY
----------
3090  IF PEEK ( - 16384) = 212 THEN TD = 1
  * : L = 2
  * : VT = 0
  * : POKE - 16368,0
  * : POP
  * : PRINT A$"2"
  * : GOTO 4180 REM DISPLAY TWO SETS OF TIMING MARKS
----------
3100  IF PEEK ( - 16384) = 194 THEN TD = 3
  * : L = 1
  * : VT = 1
  * : POKE - 16368,0
  * : POP
  * : PRINT A$"2"
  * : GOTO 4180
  * : REM DISPLAY BUS TIMING MARKS
----------
3110  IF PEEK ( - 16384) = 193 THEN TD = 2
  * : L = 1
  * : VT = 1
  * : POKE - 16368,0
  * : POP
  * : PRINT A$"2"
  * : GOTO 4180
```

```
  * : REM DISPLAY AUTO TIMING MARKS
---------------------
3120  IF PEEK ( - 16384) = 213 THEN TD = 4
  * : L = 1
  * : VT = 0
  * : POKE - 16368,0
  * : POP
  * : PRINT A$"2"
  * : GOTO 4180
  * : REM  DISPLAY ONE LINE FOR BOTH VEHICLES
---------------------
3130  IF PEEK ( - 16384) = 195 THEN POP
  * : GOTO 4000
  * : REM CHANGE PARAMETERS
---------------------
3140  IF PEEK ( - 16384) = 205 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN MENU"
---------------------
3150  IF PEEK ( - 16384) = 200 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN POISSON HELP"
---------------------
3160  IF PEEK ( - 16384) = 209 THEN POKE - 16368,0
  * : PRINT Y$;P$;Z$
  * : END
---------------------
3170  CS = PEEK (115) + PEEK (116) * 256
---------------------
3180  IF PEEK ( - 16384) = 178 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD QUEUE.SET,A",CS
  * : PRINT CHR$ (4);"RUN MM1Q"
---------------------
3190  IF PEEK ( - 16384) = 179 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD DAM.SET,A",CS
  * : PRINT CHR$ (4);"RUN DAM MODEL"
---------------------
3200  IF PEEK ( - 16384) = 180 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD DAMAGE.SET,A",CS
  * : PRINT CHR$ (4);"RUN DAMAGE MODEL"
---------------------
3210  IF PEEK ( - 16384) = 181 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD FISH.SET,A",CS
  * : PRINT CHR$ (4);"RUN FISHPOND"
---------------------
3220  IF PEEK ( - 16384) < > 160 THEN 3250
---------------------
3230  IF PEEK ( - 16384) = 160 THEN POKE - 16368,0
  * : REM CLOCK STOPPED TO PAUSE
---------------------
3240  IF PEEK ( - 16384) < > 160 THEN GOTO 3240
  * : REM AWAITING SPC BAR TO RESUME
---------------------
```

```
3250  POKE - 16368,0
    : POKE 36,CH
    : POKE 37,CV
    : RETURN
----------------------
3499  REM DISPLAY SET-UP SUBROUTINE
----------------------
3500  PRINT Y$;A$"2"
    : VTAB 18
    : HTAB 5
    : PRINT S$;"TIME= ";TT
    : VTAB 19
    : HTAB 1
    : PRINT S$;"COUNT( )= ";C1
    : VTAB 19
    : HTAB 17
    : PRINT "    =";
    : PRINT FOR 5,2;#G1
----------------------
3510  VTAB 19
    : HTAB 31
    : PRINT "    =";
    : PRINT FOR 5,2;#M1
    : VTAB 20
    : HTAB 1
    : PRINT S$;"COUNT( )= ";C2
    : VTAB 20
    : HTAB 17
    : PRINT "    =";
    : PRINT FOR 5,2;#G2
    : VTAB 20
    : HTAB 31
    : PRINT "    =";
    : PRINT FOR 5,2;#M2
----------------------
3520  PRINT K$;A$"1"
    : VTAB 19
    : HTAB 7
    : PRINT "5"
    : VTAB 19
    : HTAB 18
    : PRINT "45"
    : VTAB 19
    : HTAB 32
    : PRINT "75"
    : VTAB 20
    : HTAB 7
    : PRINT "6"
    : VTAB 20
    : HTAB 18
    : PRINT "46"
```

```
    : VTAB 20
    : HTAB 32
    : PRINT "76"
    : RETURN
---------------------
3999  REM PARAMETER CHANGE SUBROUTINE
---------------------
4000  PRINT A$"2";P$
    : POKE - 16368,0
---------------------
4010  VTAB 6
    : HTAB 1
    : PRINT S$;"PRESENT MEANS;";C$;S$;"INTERARRIVAL ";S$;"TI
        ME (AUTO)";C$;S$;"INTERARRIVAL ";S$;"TIME (BUS)"
---------------------
4020  VTAB 7
    : HTAB 26
    : PRINT FOR 5,2;#G1
    : VTAB 8
    : HTAB 26
    : PRINT FOR 5,2;#G2
---------------------
4030  VTAB 10
    : HTAB 1
    : PRINT S$;"NEW MEANS ? ";
    : GET R$
    : IF R$ = "Y" THEN 4060
---------------------
4040  IF R$ = "N" THEN 4170
---------------------
4050  IF R$ < > "N" THEN 4030
---------------------
4060  VTAB 12
    : HTAB 1
    : PRINT F$;S$;"INTERARRIVAL ";S$;"TIME (AUTO) ";
    : INPUT "";R$
---------------------
4070  IF VAL (R$) = 0 THEN 4110
---------------------
4080  IF VAL (R$) < = 1 THEN GOSUB 4500
  * : GOTO 4060
---------------------
4090  IF VAL (R$) > 99 THEN GOSUB 4510
  * : GOTO 4060
---------------------
4100  G1 = VAL (R$)
---------------------
4110  VTAB 12
    : HTAB 1
    : PRINT F$;S$;"INTERARRIVAL ";S$;"TIME (BUS)";
    : INPUT "";R$
```

```
--------------------
4120  IF VAL (R$) = 0 THEN 4160
--------------------
4130  IF VAL (R$) < = 1 THEN GOSUB 4500
  * : GOTO 4110
--------------------
4140  IF VAL (R$) > 99 THEN GOSUB 4510
  * : GOTO 4110
--------------------
4150  G2 = VAL (R$)
--------------------
4160  VTAB 12
    : HTAB 1
    : PRINT F$
    : GOTO 4020
--------------------
4170  E1 = - 1 / LOG (1 - 1 / G1)
    : E2 = - 1 / LOG (1 - 1 / G2)
    : P = E1 / (E1 + E2)
--------------------
4180  PRINT P$;Y$
--------------------
4190  IF SD = 1 THEN VTAB 18
  * : HTAB 11
  * : PRINT "0      "
  * : VTAB 19
  * : HTAB 11
  * : PRINT "0      "
  * : VTAB 19
  * : HTAB 21
  * : PRINT FOR 5,2;#G1
  * : VTAB 19
  * : HTAB 35
  * : PRINT " 0.00"
  * : VTAB 20
  * : HTAB 11
  * : PRINT "0      "
  * : VTAB 20
  * : HTAB 21
  * : PRINT FOR 5,2;#G2
  * : VTAB 20
  * : HTAB 35
  * : PRINT " 0.00"
--------------------
4200  GOSUB 10210
    : PRINT K$;A$"1"
--------------------
4210  IF TD = 1 THEN VTAB 3
  * : HTAB 1
  * : GOTO 1000
--------------------
```

```
4220  VTAB 6
    : HTAB 1
    : GOTO 1000
----------------------
4500  VTAB 12
    : HTAB 1
    : PRINT F$;S$;"(VALUE MUST BE > 1)"
    : FOR I = 1 TO 2000
    : NEXT
    : RETURN
----------------------
4510  VTAB 12
    : HTAB 1
    : PRINT F$;S$;"(INPUT VALUES < 100)"
    : FOR I = 1 TO 2000
    : NEXT
    : RETURN
----------------------
9999  REM INITIALIZE SUBROUTINE
----------------------
10000 A$ = CHR$ (1)
    : B$ = CHR$ (2)
    : C$ = CHR$ (3)
    : D$ = CHR$ (4)
    : E$ = CHR$ (5)
    : F$ = CHR$ (6)
    : K$ = CHR$ (11)
    : L$ = CHR$ (12)
    : O$ = CHR$ (15)
    : P$ = CHR$ (16)
    : Q$ = CHR$ (17)
    : S$ = CHR$ (19)
    : V$ = CHR$ (22)
    : W$ = CHR$ (23)
    : Y$ = CHR$ (25)
    : Z$ = CHR$ (26)
----------------------
10010 DIM CAR$(3),DOT$(3),DASH$(3),MIN$(3),DIAM$(9)
----------------------
10020 CAR$(1) = B$ + "  ABC" + C$ + " DEFGHI" + Q$ + D$
----------------------
10030 CAR$(2) = B$ + " KLMNMPQ" + C$ + " RSTUVSW" + Q$ + D$
----------------------
10040 DOT$(1) = "X"
----------------------
10050 DOT$(2) = B$ + "X" + C$ + "X" + Q$ + D$
----------------------
10060 DASH$(1) = "Y"
----------------------
10070 DASH$(2) = B$ + "Y" + C$ + "Y" + Q$ + D$
----------------------
```

```
10080 MIN$(1) = "Z"
----------
10090 MIN$(2) = B$ + "Z" + C$ + "Z" + Q$ + D$
----------
10100 DIAM$(1) = "1"
----------
10110 DIAM$(2) = B$ + "X" + C$ + "2" + Q$ + D$
----------
10120 DIAM$(3) = B$ + "Y" + C$ + "2" + Q$ + D$
----------
10130 DIAM$(4) = B$ + "Z" + C$ + "2" + Q$ + D$
----------
10140 DIAM$(5) = B$ + "1" + C$ + "X" + Q$ + D$
----------
10150 DIAM$(6) = B$ + "1" + C$ + "Y" + Q$ + D$
----------
10160 DIAM$(7) = B$ + "1" + C$ + "Z" + Q$ + D$
----------
10170 DIAM$(8) = "2"
----------
10180 POKE - 16368,0
    : PRINT Z$;O$;A$;O$;D$;P$;A$"1"
----------
10190 HCOLOR= 3
    : HPLOT 1,32 TO 279,32
----------
10200 E1 = 4.48142012
    : E2 = 5.48481495
    : G1 = 5
    : G2 = 6
    : L = 1
    : N = 1
    : NN = 1
    : P = E1 / (E1 + E2)
    : R = 1000
    : SD = 0
    : SPEED= 255
    : TD = 4
    : VT = 0
----------
10210 C1 = 0
    : C2 = 0
    : M1 = 0
    : M2 = 0
    : TM = 0
    : TT = 0
----------
10220 GOSUB 2500
    : VTAB 6
    : HTAB 1
    : REM  INITIALIZE FIRST EVENT
```

```
----------------------
10230 RETURN
```

B. MM1Q PROGRAM

```
---------------------
10    GOSUB 10000
    : REM INITIALIZE
---------------------
999   REM MAIN ROUTINE;CLOCK
---------------------
1000  IF TM = 30 OR TM = 0 THEN POKE 37,( PEEK (37) + 1)
  * : PRINT C$
---------------------
1010  IF PEEK (37) > 15 THEN POKE 37,15
---------------------
1020  IF T1 = T2 AND T1 = 0 THEN EVT = 1
  * : GOSUB 1500
  * : GOTO 1100
---------------------
1030  IF SYS < > 0 AND T1 = 0 THEN EVT = 2
  * : GOSUB 1500
  * : GOTO 1100
  * : REM A DEPARTURE EVENT
---------------------
1040  IF T2 = 0 THEN EVT = 3
  * : GOSUB 1500
  * : GOTO 1100
  * : REM AN ARRIVAL EVENT
---------------------
1050  IF NN = 0 THEN 1090
---------------------
1060  IF TM / 60 = INT (TM / 60) THEN PRINT MIN$(L);
  * : GOTO 1090
---------------------
1070  IF TM / 5 = INT (TM / 5) THEN PRINT DASH$(L);
  * : GOTO 1090
---------------------
1080  PRINT DOT$(L);
---------------------
1090  FOR I = 1 TO R
    : NEXT
    : POKE 36,( PEEK (36) + 1)
---------------------
1100  T2 = T2 - 1
    : IF SYS < > 0 THEN T1 = T1 - 1
  * : REM IF NO ONE IS IN SYSTEM THEN CAN'T HAVE A DEPARTURE
        !
---------------------
1110  TM = TM + 1
    : IF TM = 60 THEN TM = 0
---------------------
1120  IF PEEK ( - 16384) > 127 THEN GOSUB 3000
  * : REM KEY PRESSED TO STOP CLOCK
```

```
---------------------
1130  TT = TT + 1
    : QT = QT + QL
    : AQL = QT / TT
    : IF QT > 32765 THEN PRINT Y$;P$
  * : GOTO 10
---------------------
1140  IF SD = 1 THEN CH = PEEK (36)
  * : CV = PEEK (37)
  * : PRINT Y$;A$"2"
  * : VTAB 19
  * : HTAB 7
  * : PRINT TT
  * : VTAB 20
  * : HTAB 35
  * : PRINT FOR 5,2;#AQL;A$"1"
  * : GOSUB 2760
  * : POKE 36,( PEEK (36) - 1)
---------------------
1150  GOTO 1000
---------------------
1500  IF NN = 0 THEN 2000
---------------------
1510  IF EVT = 1 THEN PRINT DIAM$(1);
  * : GOTO 2000
---------------------
1520  K = EVT
    : IF EVT = 3 THEN K = 5
---------------------
1530  IF TM / 60 = INT (TM / 60) THEN PRINT DIAM$(K + 2);
  * : GOTO 2000
---------------------
1540  IF TM / 5 = INT (TM / 5) THEN PRINT DIAM$(K + 1);
  * : GOTO 2000
---------------------
1550  PRINT DIAM$(K);
---------------------
2000  CH = PEEK (36)
    : CV = PEEK (37)
    : PRINT Y$
    : IF EVT = 2 THEN 2400
---------------------
2099  REM ARRIVAL EVENT ROUTINE
---------------------
2100  L = 2
    : IF SQL > 25 AND EVT = 1 THEN SYS = SYS + 1
  * : GOTO 2400
---------------------
2110  IF SQL > 25 THEN SYS = SYS + 1
  * : GOTO 2600
---------------------
```

```
2120  IF N = 0 THEN 2160
---------------------
2130  X = 1
    : DIST = 27 - SQL
    : IF SYS = 0 THEN DIST = 29 - SQL
---------------------
2140  GOSUB 2800
---------------------
2150  VTAB 3
    : HTAB (X)
    : PRINT BLNK$;
    : VTAB 3
    : HTAB (X + 1)
    : PRINT MAN$(4);
---------------------
2160  SQL = SQL + 2
    : SYS = SYS + 1
---------------------
2170  IF EVT = 3 THEN 2600
---------------------
2399  REM DEPARTURE EVENT ROUTINE
---------------------
2400  IF N = 0 THEN 2430
---------------------
2410  X = 29
    : DIST = 38
---------------------
2420  GOSUB 2800
    : VTAB 3
    : HTAB 37
    : PRINT CLR$;
---------------------
2430  SYS = SYS - 1
    : IF SYS = 0 THEN L = 1
---------------------
2440  IF SYS < 13 THEN SQL = SQL - 2
---------------------
2450  IF N = 0 OR SQL = 0 THEN 2600
---------------------
2460  VTAB 3
    : HTAB 26
    : PRINT BLNK$
    : X = 26
    : DIST = 29
    : GOSUB 2800
    : VTAB 3
    : HTAB 29
    : PRINT BLNK$
    : VTAB 3
    : HTAB 30
    : PRINT MAN$(4)
```

```
---------------------
2470  IF SQL = 2 THEN 2600
---------------------
2480  X = 23
---------------------
2490  FOR I = 2 TO SQL / 2
    : IF X < 1 THEN 2600
---------------------
2500  VTAB 3
    : HTAB (X)
    : PRINT MAN$(1);
    : FOR J = 1 TO 50
    : NEXT
    : VTAB 3
    : HTAB (X + 3)
    : PRINT MAN$(4);
    : VTAB 3
    : HTAB (X)
    : PRINT BLNK$;
    : X = X - 2
    : NEXT
---------------------
2600  QL = SYS - 1
    : IF SYS = 0 THEN QL = 0
---------------------
2610  IF SD = 1 THEN VTAB 20
  * : HTAB 19
  * : PRINT A$"2";QL;"     ";A$"1"
---------------------
2699  REM NEXT EVENT SUBROUTINE
---------------------
2700  IF EVT = 3 THEN 2740
---------------------
2710  T = - E1 * LOG ( RND (1))
---------------------
2720  T1 = INT (T) + 1
---------------------
2730  IF EVT = 2 THEN 2760
---------------------
2740  T = - E2 * LOG ( RND (1))
---------------------
2750  T2 = INT (T) + 1
---------------------
2760  VTAB 7
    : HTAB 5
    : PRINT V$;
    : VTAB 17
    : HTAB 30
    : PRINT W$;
    : POKE 36,CH
    : POKE 37,CV
```

```
     : POKE 36,( PEEK (36) + 1)
     : RETURN
---------------------
2800  I = 1
     : GOSUB 2820
     : I = 2
     : GOSUB 2820
     : X = X + 1
     : I = 3
     : GOSUB 2820
     : IF X < DIST THEN 2800
---------------------
2810  RETURN
---------------------
2820  VTAB 3
     : HTAB (X)
     : PRINT MAN$(I);
     : FOR J = 1 TO 2
     : NEXT
     : RETURN
---------------------
2999  REM KEYPRESS SUBROUTINE
---------------------
3000  CH = PEEK (36)
     : CV = PEEK (37)
---------------------
3010  IF PEEK ( - 16384) = 211 THEN R = 1000
   * : GOTO 3210
   * : REM RUN CLOCK SLOW
---------------------
3020  IF PEEK ( - 16384) = 198 THEN R = 1
   * : GOTO 3210
   * : REM RUN CLOCK FAST
---------------------
3030  IF PEEK ( - 16384) = 206 AND N = 0 AND SD = 1 THEN NN =
         0
   * : PRINT P$
   * : GOTO 3210
   * : REM SHOW STATS DISPLAY ONLY
---------------------
3040  IF PEEK ( - 16384) = 206 THEN N = 0
   * : GOTO 3210
   * : REM SHOW ALL BUT CHARACTER ANIMATION
---------------------
3050  IF PEEK ( - 16384) = 217 AND NN = 0 THEN NN = 1
   * : TM = 0
   * : CV = 3
   * : CH = 0
   * : GOTO 3210
   * : REM IN DISPLAY ONLY MODE; BRING BACK TIMING PLOT
---------------------
```

```
3060  IF PEEK ( - 16384) = 217 THEN N = 1
  * : GOTO 3600
  * : REM NOT SHOWING CHARACTER ANIMATION; BRING IT BACK
---------------------
3070  IF PEEK ( - 16384) = 196 THEN SD = 1
  * : GOSUB 3500
  * : GOSUB 2760
  * : GOTO 3210
  * : REM SET DISPLAY MODE
---------------------
3080  IF PEEK ( - 16384) = 207 AND NN = 1 THEN SD = 0
  * : PRINT Z$
  * : VTAB 18
  * : HTAB 1
  * : PRINT F$;O$;T$;A$"1"
  * : GOSUB 2760
  * : GOTO 3210
  * : REM TURN OFF DISPLAY
---------------------
3090  IF PEEK ( - 16384) = 195 THEN POP
  * : GOTO 4000
  * : REM CHANGE PARAMETERS
---------------------
3100  IF PEEK ( - 16384) = 205 THEN PRINT Z$;O$;B$
  * : VTAB 4
  * : HTAB 32
  * : PRINT CLR$;Z$
  * : PRINT CHR$ (4);"RUN MENU"
---------------------
3110  IF PEEK ( - 16384) = 200 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN QUEUE HELP"
---------------------
3120  IF PEEK ( - 16384) = 209 THEN POKE - 16368,0
  * : PRINT Y$;O$;B$
  * : VTAB 4
  * : HTAB 32
  * : PRINT CLR$;Z$;P$
  * : END
---------------------
3130  CS = PEEK (115) + PEEK (116) * 256
---------------------
3140  IF PEEK ( - 16384) = 177 THEN PRINT Z$;O$;B$
  * : VTAB 4
  * : HTAB 32
  * : PRINT CLR$;Z$
  * : PRINT CHR$ (4);"BLOAD POISSON.SET,A",CS
  * : PRINT CHR$ (4);"RUN POISSON PROCESS"
---------------------
3150  IF PEEK ( - 16384) = 179 THEN PRINT Z$;O$;B$
  * : VTAB 4
  * : HTAB 32
```

```
  * : PRINT CLR$;Z$
  * : PRINT CHR$ (4);"BLOAD DAM.SET,A",CS
  * : PRINT CHR$ (4);"RUN DAM MODEL"
---------------------
3160  IF PEEK ( - 16384) = 180 THEN PRINT Z$;O$;B$
  * : VTAB 4
  * : HTAB 32
  * : PRINT CLR$;Z$
  * : PRINT CHR$ (4);"BLOAD DAMAGE.SET,A",CS
  * : PRINT CHR$ (4);"RUN DAMAGE MODEL"
---------------------
3170  IF PEEK ( - 16384) = 181 THEN PRINT Z$;O$;B$
  * : VTAB 4
  * : HTAB 32
  * : PRINT CLR$;Z$
  * : PRINT CHR$ (4);"BLOAD FISH.SET,A",CS
  * : PRINT CHR$ (4);"RUN FISHPOND"
---------------------
3180  IF PEEK ( - 16384) < > 160 THEN 3210
---------------------
3190  IF PEEK ( - 16384) = 160 THEN POKE - 16368,0
  * : REM CLOCK STOPPED TO PAUSE
---------------------
3200  IF PEEK ( - 16384) < > 160 THEN 3200
  * : REM AWAITING SPC BAR TO RESUME
---------------------
3210  POKE - 16368,0
    : POKE 36,CH
    : POKE 37,CV
    : RETURN
---------------------
3499  REM DISPLAY SET-UP SUBROUTINE
---------------------
3500  PRINT Y$
    : VTAB 19
    : HTAB 1
    : PRINT A$"2";S$;"TIME= ";TT
    : VTAB 20
    : HTAB 15
    : PRINT S$;"Q#= ";QL
    : VTAB 20
    : HTAB 27
    : PRINT S$;"AVG ";S$;"Q#= ";K$;
    : PRINT FOR 5,2;#AQL;A$"1"
    : RETURN
---------------------
3599  REM CLEANUP SUBROUTINE
---------------------
3600  PRINT Y$;
    : FOR I = 1 TO 30 STEP 2
    : VTAB 3
```

```
    : HTAB (I)
    : PRINT BLNK$;
    : NEXT
----------------------
3610  IF SQL = 0 THEN 3640
----------------------
3620  VTAB 3
    : HTAB 30
    : PRINT MAN$(4)
----------------------
3630  IF SQL > 2 THEN X = 26
  * : FOR I = 4 TO SQL STEP 2
  * : VTAB 3
  * : HTAB (X)
  * : PRINT MAN$(4);
  * : X = X - 2
  * : NEXT
----------------------
3640  POKE - 16368,0
    : VTAB 7
    : HTAB 5
    : PRINT V$;
    : VTAB 17
    : HTAB 30
    : PRINT W$;
    : POKE 36,CH
    : POKE 37,CV
    : RETURN
----------------------
3999  REM PARAMETER CHANGE SUBROUTINE
----------------------
4000  PRINT A$"2";P$
    : POKE - 16368,0
----------------------
4010  VTAB 8
    : HTAB 1
    : PRINT S$;"PRESENT ";S$;"MEANS;";C$;S$;"SERVICE ";S$;"T
        IME";C$;S$;"INTERARRIVAL ";S$;"TIME"
----------------------
4020  VTAB 9
    : HTAB 19
    : PRINT FOR 5,2;#G1
    : VTAB 10
    : HTAB 19
    : PRINT FOR 5,2;#G2
----------------------
4030  VTAB 12
    : HTAB 1
    : PRINT S$;"NEW ";S$;"MEANS ? ";K$;
    : GET R$
    : IF R$ = "Y" THEN 4060
```

```
--------------------
4040  IF R$ = "N" THEN 4170
--------------------
4050  GOTO 4030
--------------------
4060  VTAB 14
    : HTAB 1
    : PRINT S$;"SERVICE ";S$;"TIME ";
    : INPUT "";R$
--------------------
4070  IF VAL (R$) = 0 THEN 4110
--------------------
4080  IF VAL (R$) < = 1 THEN GOSUB 4500
  * : GOTO 4060
--------------------
4090  IF VAL (R$) > 99 THEN GOSUB 4510
  * : GOTO 4060
--------------------
4100  G1 = VAL (R$)
--------------------
4110  VTAB 14
    : HTAB 1
    : PRINT F$;S$;"INTERARRIVAL ";S$;"TIME ";
    : INPUT "";R$
--------------------
4120  IF VAL (R$) = 0 THEN 4160
--------------------
4130  IF VAL (R$) < = 1 THEN GOSUB 4500
  * : GOTO 4110
--------------------
4140  IF VAL (R$) > 99 THEN GOSUB 4510
  * : GOTO 4110
--------------------
4150  G2 = VAL (R$)
--------------------
4160  VTAB 12
    : HTAB 1
    : PRINT F$
    : GOTO 4020
--------------------
4170  E1 = - 1 / LOG (1 - 1 / G1)
    : E2 = - 1 / LOG (1 - 1 / G2)
--------------------
4180  PRINT P$;Y$
    : IF NN = 1 THEN FOR I = 1 TO 38 STEP 2
  * : VTAB 3
  * : HTAB (I)
  * : PRINT BLNK$
  * : NEXT
--------------------
4190  IF SD = 1 THEN VTAB 19
```

```
   * : HTAB 7
   * : PRINT "Ø      "
   * : VTAB 2Ø
   * : HTAB 19
   * : PRINT "Ø    "
   * : VTAB 2Ø
   * : HTAB 35
   * : PRINT " Ø.ØØ"
----------------------
4200  IF NN = 1 THEN PRINT A$"1"
   * : GOSUB 1Ø25Ø
   * : CH = 1
   * : CV = 7
   * : GOTO 1ØØØ
----------------------
4210  GOSUB 1Ø25Ø
     : CH = 1
     : CV = 7
     : GOTO 1ØØØ
----------------------
4500  VTAB 14
     : HTAB 1
     : PRINT F$;S$;"(VALUE MUST BE > 1)"
     : FOR I = 1 TO 2ØØØ
     : NEXT
     : VTAB 14
     : HTAB 1
     : PRINT F$
     : RETURN
----------------------
4510  VTAB 14
     : HTAB 1
     : PRINT F$;S$;"(INPUT VALUES < 1ØØ)"
     : FOR I = 1 TO 2ØØØ
     : NEXT
     : VTAB 14
     : HTAB 1
     : PRINT F$
     : RETURN
----------------------
9999  REM INITIALIZE SUBROUTINE
----------------------
1ØØØØ A$ = CHR$ (1)
     : B$ = CHR$ (2)
     : C$ = CHR$ (3)
     : D$ = CHR$ (4)
     : E$ = CHR$ (5)
     : F$ = CHR$ (6)
     : K$ = CHR$ (11)
     : L$ = CHR$ (12)
     : O$ = CHR$ (15)
```

```
    : P$ = CHR$ (16)
    : Q$ = CHR$ (17)
    : S$ = CHR$ (19)
    : T$ = CHR$ (20)
    : V$ = CHR$ (22)
    : W$ = CHR$ (23)
    : Y$ = CHR$ (25)
    : Z$ = CHR$ (26)
----------
10010 DIM MAN$(5),DOT$(2),DASH$(2),MIN$(2),DIAM$(7)
----------
10020 MAN$(1) = B$ + " A" + C$ + " BC" + C$ + " DE" + D$
----------
10030 MAN$(2) = B$ + " FG" + C$ + " HI" + C$ + " JK" + D$
----------
10040 MAN$(3) = B$ + "LM " + C$ + "NOP" + C$ + "QR " + D$
----------
10050 MAN$(4) = B$ + "S" + C$ + "T" + C$ + "U" + D$
----------
10060 MAN$(5) = B$ + "22" + C$ + "22" + D$
----------
10070 DOT$(1) = B$ + "V" + C$ + " " + Q$ + D$
----------
10080 DOT$(2) = B$ + "V" + C$ + "V" + Q$ + D$
----------
10090 DASH$(1) = B$ + "W" + C$ + " " + Q$ + D$
----------
10100 DASH$(2) = B$ + "W" + C$ + "W" + Q$ + D$
----------
10110 MIN$(1) = B$ + "X" + C$ + " " + Q$ + D$
----------
10120 MIN$(2) = B$ + "X" + C$ + "X" + Q$ + D$
----------
10130 DIAM$(1) = B$ + "Y" + C$ + "1" + Q$ + D$
----------
10140 DIAM$(2) = B$ + "V" + C$ + "1" + Q$ + D$
----------
10150 DIAM$(3) = B$ + "W" + C$ + "1" + Q$ + D$
----------
10160 DIAM$(4) = B$ + "X" + C$ + "1" + Q$ + D$
----------
10170 DIAM$(5) = B$ + "Y" + C$ + "V" + Q$ + D$
----------
10180 DIAM$(6) = B$ + "Y" + C$ + "W" + Q$ + D$
----------
10190 DIAM$(7) = B$ + "Y" + C$ + "X" + Q$ + D$
----------
10200 CLR$ = B$ + "    " + C$ + "    " + C$ + "    " + D$
----------
10210 BLNK$ = B$ + "   " + C$ + "   " + C$ + "   " + D$
----------
```

```
10220 POKE - 16368,0
---------------------
10230 PRINT Z$;O$;B$;A$"1"
    : FOR I = 1 TO 38 STEP 2
    : VTAB 4
    : HTAB I
    : PRINT CLR$;
    : NEXT
    : VTAB 4
    : HTAB 32
    : PRINT MAN$(5);O$;A$;O$;D$;P$
    : VTAB 4
    : HTAB 32
    : PRINT MAN$(5);O$;T$
    : HCOLOR= 3
    : HPLOT 1,40 TO 279,40
---------------------
10240 E1 = 4.48142012
    : E2 = 5.48481495
    : G1 = 5
    : G2 = 6
    : N = 1
    : NN = 1
    : R = 1000
    : SPEED= 255
---------------------
10250 AQL = 0
    : EVT = 3
    : L = 1
    : QL = 0
    : QT = 0
    : SQL = 0
    : SYS = 0
    : TM = 0
    : TT = 0
---------------------
10260 GOSUB 2700
    : VTAB 4
    : HTAB 1
    : REM INITIALIZE FIRST EVENT
---------------------
10270 RETURN
```

C. DAM MODEL PROGRAM

```
---------------------
10    GOSUB 10000
    : REM INITIALIZE
---------------------
20    GOTO 2000
---------------------
999   REM WATER PLOT ALGORITHM
---------------------
1000  Z = .159 - (Y - 97) * .00258
    : X = INT (279 - 82.8 * (( - LOG (6.28 * Z)) ^ .5))
    : IF X = 2 * ( INT (X / 2)) THEN X = X + 1
---------------------
1010  RETURN
---------------------
1999  REM OUTFLOW PLOT ROUTINE
---------------------
2000  IF PEEK ( - 16384) > 127 THEN GOSUB 3000
  * : REM KEY PRESSED TO STOP
---------------------
2010  HCOLOR= 0
    : IF UP = 1 THEN 2300
---------------------
2020  FOR I = 0 TO T2 - 1
---------------------
2030  GOSUB 1000
---------------------
2040  IF Y > 156 THEN FOR J = 1 TO R
  * : NEXT
  * : GOTO 2140
---------------------
2049  REM OUTFLOW SIMULATION
---------------------
2050  FOR J = 1 TO R
    : NEXT
    : IF Y < = 150 THEN HPLOT 1,156 TO 34,150
  * : HCOLOR= 6
  * : HPLOT 1,156 TO 34,150
  * : HCOLOR= 0
---------------------
2059  REM RECESSION OF OVERFLOW
---------------------
2060  IF Y < 115 THEN HPLOT (45 + (114 - Y) ^ 2),Y TO X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 0
  * : Y = Y + 1
  * : LVL = 0
---------------------
2070  IF Y = 113 THEN HPLOT 47,114 TO 37,150
```

```
  * : HPLOT 46,114 TO 36,150
  * : HPLOT 35,151 TO 2,156
----------
2080  IF Y = 111 THEN HPLOT 45,114 TO 35,150
----------
2089  REM NORMAL OUTFLOW
----------
2090  IF Y < 130 AND Y > = 115 AND LVL = 3 THEN HPLOT 61,Y TO
        X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 0
  * : LVL = 0
  * : Y = Y + 1
----------
2100  IF Y < 143 AND Y > = 130 AND LVL = 2 THEN HPLOT 61,Y TO
        X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 0
  * : LVL = 0
  * : Y = Y + 1
----------
2110  IF Y > 150 THEN HPLOT 1,Y TO 34,Y
----------
2120  IF Y > = 143 THEN HPLOT 61,Y TO X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 0
  * : LVL = 0
  * : Y = Y + 1
----------
2130  LVL = LVL + 1
----------
2140  NEXT
----------
2150  IF DWN = 1 THEN 2000
----------
2160  IF PEEK ( - 16384) > 127 THEN GOSUB 3000
  * : REM KEY PRESSED TO STOP
----------
2199  REM INFLOW PLOT ROUTINE
----------
2200  IF UP = 1 THEN 2300
----------
2209  REM MOVE CLOUD IN AND RAIN
----------
2210  FOR I = 1 TO 15
    : VTAB 1
    : HTAB 1
    : PRINT CLOUD$(I)
```

```
     : NEXT
---------------------
2220  FOR I = 1 TO 15
     : VTAB 1
     : HTAB I
     : PRINT CLOUD$(15)
     : NEXT
---------------------
2230  IF T1 < 5 THEN 2260
---------------------
2240  FOR I = 1 TO 3
     : VTAB 5
     : HTAB 21
     : PRINT LIGHT$
     : VTAB 5
     : HTAB 21
     : PRINT LBLNK$
     : NEXT
     : VTAB 5
     : HTAB 21
     : PRINT LIGHT$
     : FOR I = 1 TO 800
     : NEXT
     : VTAB 5
     : HTAB 21
     : PRINT LBLNK$
---------------------
2250  IF T1 > 15 THEN FOR I = 1 TO 800
   * : NEXT
   * : FOR I = 1 TO 3
   * : VTAB 5
   * : HTAB 21
   * : PRINT LIGHT$
   * : VTAB 5
   * : HTAB 21
   * : PRINT LBLNK$
   * : NEXT
   * : VTAB 5
   * : HTAB 21
   * : PRINT LIGHT$
   * : FOR I = 1 TO 800
   * : NEXT
   * : VTAB 5
   * : HTAB 21
   * : PRINT LBLNK$
---------------------
2260  FOR I = 1 TO 800
     : NEXT
     : VTAB 5
     : HTAB 16
     : PRINT RAIN$
```

```
----------
2299  REM INFLOW ROUTINE
----------
2300  HCOLOR= 6
----------
2310  FOR I = 0 TO T1 - 1
----------
2320  GOSUB 1000
----------
2330  FOR J = 1 TO R
    : NEXT
----------
2340  IF Y < 108 THEN 2430
----------
2349  REM OVERFLOW SIMULATION
----------
2350  IF Y < 115 THEN HPLOT (45 + (114 - Y) ^ 2),Y TO X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 6
  * : Y = Y - 1
  * : LVL = 0
----------
2360  IF Y = 113 THEN HPLOT 47,114 TO 37,150
  * : HPLOT 46,114 TO 36,150
  * : HPLOT 35,151 TO 2,156
----------
2370  IF Y = 111 THEN HPLOT 45,114 TO 35,150
----------
2379  REM NORMAL INFLOW SIMULATION
----------
2380  IF Y < 130 AND Y > = 115 AND LVL = 3 THEN HPLOT 61,Y TO
        X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 6
  * : LVL = 0
  * : Y = Y - 1
----------
2390  IF Y < 143 AND Y > = 130 AND LVL = 2 THEN HPLOT 61,Y TO
        X,Y
  * : HCOLOR= 5
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 6
  * : LVL = 0
  * : Y = Y - 1
----------
2400  IF Y > 150 THEN HPLOT (157 - Y) * 5,Y TO 34,Y
----------
2410  IF Y > = 143 THEN HPLOT 61,Y TO X,Y
  * : HCOLOR= 5
```

```
  * : HPLOT X + 2,Y TO 279,Y
  * : HCOLOR= 6
  * : LVL = 0
  * : Y = Y - 1
----------
2420  LVL = LVL + 1
----------
2430  NEXT
----------
2440  IF UP = 1 THEN 2000
----------
2500  FOR I = 5 TO 7
    : VTAB I
    : HTAB 16
    : PRINT RBLNK$
    : FOR J = 1 TO 100
    : NEXT
    : NEXT
----------
2510  FOR I = 15 TO 26
    : VTAB 1
    : HTAB I
    : PRINT CLOUD$(15)
    : NEXT
----------
2520  FOR I = 27 TO 40
    : VTAB 1
    : HTAB I
    : PRINT CLOUD$(I - 11)
    : NEXT
----------
2799  REM SET UP FOR NEXT INFLOW
----------
2800  T = - 5 * LOG ( RND (1))
----------
2810  T1 = INT (T) + 1
----------
2820  T = - 5 * LOG ( RND (1))
----------
2830  T2 = INT (T) + 1
----------
2840  IF INIT = 1 THEN RETURN
----------
2850  GOTO 2000
----------
2999  REM CLOCK STOPPED, FIND OUT WHY
----------
3000  IF PEEK ( - 16384) = 211 THEN R = 1000
  * : GOTO 3160
  * : REM RUN CLOCK SLOW
----------
```

```
3010  IF PEEK ( - 16384) = 198 THEN R = 1
  * : GOTO 3160
  * : REM RUN CLOCK FAST
---------------------
3020  IF PEEK ( - 16384) = 213 THEN UP = 1
  * : DWN = 0
  * : GOTO 3160
  * : REM WATER INFLOW ONLY
---------------------
3030  IF PEEK ( - 16384) = 196 THEN UP = 0
  * : DWN = 1
  * : GOTO 3160
  * : REM WATER OUTLOW ONLY
---------------------
3040  IF PEEK ( - 16384) = 206 THEN UP = 0
  * : DWN = 0
  * : GOTO 3160
  * : REM RESUME NORMAL IN/OUT FLOW
---------------------
3050  IF PEEK ( - 16384) = 205 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN MENU"
---------------------
3060  IF PEEK ( - 16384) = 200 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN DAM HELP"
---------------------
3070  IF PEEK ( - 16384) = 209 THEN POKE - 16368,0
  * : PRINT Y$;P$;Z$
  * : END
---------------------
3080  CS = PEEK (115) + PEEK (116) * 256
---------------------
3090  IF PEEK ( - 16384) = 177 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD POISSON.SET,A",CS
  * : PRINT CHR$ (4);"RUN POISSON PROCESS"
---------------------
3100  IF PEEK ( - 16384) = 178 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD QUEUE.SET,A",CS
  * : PRINT CHR$ (4);"RUN MM1Q"
---------------------
3110  IF PEEK ( - 16384) = 180 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD DAMAGE.SET,A",CS
  * : PRINT CHR$ (4);"RUN DAMAGE MODEL"
---------------------
3120  IF PEEK ( - 16384) = 181 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD FISH.SET,A",CS
  * : PRINT CHR$ (4);"RUN FISHPOND"
---------------------
3130  IF PEEK ( - 16384) < > 160 THEN 3160
---------------------
3140  IF PEEK ( - 16384) = 160 THEN POKE - 16368,0
  * : REM CLOCK STOPPED TO PAUSE
```

```
----------
3150  IF PEEK ( - 16384) < > 160 THEN GOTO 3150
  * : REM AWAITING SPC BAR TO RESUME
----------
3160  POKE - 16368,0
    : RETURN
----------
9999  REM INITIALIZE SUBROUTINE
----------
10000 A$ = CHR$ (1)
    : B$ = CHR$ (2)
    : C$ = CHR$ (3)
    : D$ = CHR$ (4)
    : K$ = CHR$ (11)
    : L$ = CHR$ (12)
    : P$ = CHR$ (16)
    : Q$ = CHR$ (17)
    : S$ = CHR$ (19)
    : Y$ = CHR$ (25)
    : Z$ = CHR$ (26)
----------
10010 DIM CLOUD$(29)
----------
10020 CLOUD$(1) = B$ + " " + C$ + " " + C$ + L$ + "A" + C$ +
        "E" + K$ + D$
----------
10030 CLOUD$(2) = B$ + " " + C$ + "Q" + C$ + "Z" + L$ + "A" +
        C$ + "CE" + K$ + D$
----------
10040 CLOUD$(3) = B$ + " " + C$ + "PQ" + C$ + " Z" + L$ + "A"
        + C$ + "DCE" + K$ + D$
----------
10050 CLOUD$(4) = B$ + " " + C$ + "OPQ" + C$ + "Y Z" + L$ + "
        A" + C$ + "DDCE" + K$ + D$
----------
10060 CLOUD$(5) = B$ + "H" + C$ + "NOPQ" + C$ + "XY Z" + L$ +
        "A" + C$ + "DDDCE" + K$ + D$
----------
10070 CLOUD$(6) = B$ + "GH" + C$ + " NOPQ" + C$ + "WXY Z" + L
        $ + "A" + C$ + "DDDDCE" + K$ + D$
----------
10080 CLOUD$(7) = B$ + "FGH" + C$ + "  NOPQ" + C$ + " WXY Z"
        + L$ + "A" + C$ + "DDDDDCE" + K$ + D$
----------
10090 CLOUD$(8) = B$ + "EFGH" + C$ + "   NOPQ" + C$ + "  WXY
        Z" + L$ + "A" + C$ + "DDDDDDCE" + K$ + D$
----------
10100 CLOUD$(9) = B$ + "DEFGH" + C$ + "M   NOPQ" + C$ + "V  W
        XY Z" + L$ + "A" + C$ + "CDDDDDDCE" + K$ + D$
----------
10110 CLOUD$(10) = B$ + "CDEFGH" + C$ + "LM   NOPQ" + C$ + "
```

```
V  WXY Z" + L$ + "A" + C$ + "CCDDDDDDCE" + K$ + D$
----------
10120 CLOUD$(11) = B$ + "BCDEFGH" + C$ + "KLM   NOPQ" + C$ +
        "U V  WXY Z" + L$ + "A" + C$ + "CCCDDDDDDCE" + K$ +
        D$
----------
10130 CLOUD$(12) = B$ + "ABCDEFGH" + C$ + "JKLM   NOPQ" + C$
        + "TU V  WXY Z" + L$ + "A" + C$ + "CCCCDDDDDDCE" +
        K$ + D$
----------
10140 CLOUD$(13) = B$ + " ABCDEFGH" + C$ + "IJKLM   NOPQ" + C
        $ + "STU V  WXY Z" + L$ + "A" + C$ + "CCCCCDDDDDDCE
        " + K$ + D$
----------
10150 CLOUD$(14) = B$ + "  ABCDEFGH" + C$ + " IJKLM   NOPQ" +
        C$ + "RSTU V  WXY Z" + L$ + "A" + C$ + "BCCCCCDDDDD
        DCE" + K$ + D$
----------
10160 CLOUD$(15) = B$ + "   ABCDEFGH" + C$ + "  IJKLM   NOPQ"
        + C$ + " RSTU V  WXY Z" + L$ + "A" + C$ + " BCCCCCD
        DDDDDCE" + K$ + D$
----------
10170 CLOUD$(16) = B$ + "   ABCDEFGH" + C$ + "  IJKLM   NOPQ"
        + C$ + " RSTU V WXY Z" + L$ + C$ + " BCCCCCDDDDDDC"
        + K$ + D$
----------
10180 CLOUD$(17) = B$ + "   ABCDEFGH" + C$ + "  IJKLM   NOP"
        + C$ + " RSTU V  WXY " + L$ + C$ + " BCCCCCDDDDDD"
        + K$ + D$
----------
10190 CLOUD$(18) = B$ + "   ABCDEFGH" + C$ + "  IJKLM   NO" +
        C$ + " RSTU V  WXY" + L$ + C$ + " BCCCCCDDDDD" + K$
        + D$
----------
10200 CLOUD$(19) = B$ + "   ABCDEFGH" + C$ + "  IJKLM   N" +
        C$ + " RSTU V  WX" + L$ + C$ + " BCCCCCDDDD" + K$ +
        D$
----------
10210 CLOUD$(20) = B$ + "   ABCDEFG" + C$ + "  IJKLM   " + C$
        + " RSTU V  W" + L$ + C$ + " BCCCCCDDD" + K$ + D$
----------
10220 RAIN$ = B$ + L$ + "OFGMGFPOFGMGFP" + C$ + "QNFPFFSQNFPF
        FS" + C$ + "QQRSRQRQQRSRQR" + K$ + D$
----------
10230 CLOUD$(21) = B$ + "   ABCDEF" + C$ + "  IJKLM  " + C$ +
        " RSTU V  " + L$ + C$ + " BCCCCCDD" + K$ + D$
----------
10240 CLOUD$(22) = B$ + "   ABCDE" + C$ + "  IJKLM " + C$ + "
         RSTU V " + L$ + C$ + " BCCCCCD" + K$ + D$
----------
10250 CLOUD$(23) = B$ + "   ABCD" + C$ + "  IJKLM" + C$ + " R
```

```
STU V" + L$ + C$ + " BCCCCC" + K$ + D$
---------------------
10260 CLOUD$(24) = B$ + "   ABC" + C$ + "  IJKL" + C$ + " RST
      U " + L$ + C$ + " BCCCC" + K$ + D$
---------------------
10270 CLOUD$(25) = B$ + "   AB" + C$ + "  IJK" + C$ + " RSTU"
      + L$ + C$ + " BCCC" + K$ + D$
---------------------
10280 CLOUD$(26) = B$ + "   A" + C$ + "  IJ" + C$ + " RST" +
      L$ + C$ + " BCC" + K$ + D$
---------------------
10290 CLOUD$(27) = B$ + "    " + C$ + "  I" + C$ + " RS" + L$
      + C$ + " BC" + K$ + D$
---------------------
10300 CLOUD$(28) = B$ + "  " + C$ + "  " + C$ + " R" + L$ + C
      $ + " B" + K$ + D$
---------------------
10310 CLOUD$(29) = B$ + " " + C$ + " " + C$ + " " + C$ + " "
      + D$
---------------------
10320 RAIN$ = B$ + L$ + "OFGMGFPOFGMGFP" + C$ + "QNFPFFSQNFPF
      FS" + C$ + "QQRSRQRQQRSRQR" + K$ + D$
---------------------
10330 LIGHT$ = B$ + L$ + "HI" + C$ + "JK" + C$ + "L" + K$ + D
      $
---------------------
10340 RBLNK$ = B$ + "                " + D$
---------------------
10350 LBLNK$ = B$ + "  " + C$ + "  " + C$ + " " + D$
---------------------
10360 POKE - 16368,0
    : PRINT Z$;O$;A$;O$;D$;P$;A$,"2"
    : VTAB 3
    : HTAB 1
    : PRINT S$;"ONE ";S$;"MOMENT ";S$;"PLEASE...";K$
---------------------
10370 PRINT CHR$ (4);"BLOAD DAM.PIC,A$2000"
---------------------
10380 PRINT A$"1"
---------------------
10390 LVL = 0
    : R = 750
    : SPEED= 255
    : Y = 139
---------------------
10400 INIT = 1
    : GOSUB 2800
    : INIT = 0
---------------------
10410 RETURN
```

D. DAMAGE MODEL PROGRAM

```
---------------------
10    GOSUB 10000
    : REM INITIALIZE
---------------------
999   REM MAIN ROUTINE;CLOCK
---------------------
1000  IF T = T1 THEN Y = Y - INC
  * : IF Y < = 55 THEN
  * : GOSUB 2000
  * : X = 14
  * : Y = 151
  * : GOTO 1000
---------------------
1010  IF T = T1 AND Y > 55 THEN GOSUB 2000
  * : GOTO 1000
---------------------
1020  HPLOT X,Y
    : X = X + 1
    : IF X > 265 THEN X = 14
---------------------
1030  IF PEEK ( - 16384) > 127 THEN GOSUB 3000
  * : REM KEY PRESSED TO STOP CLOCK
---------------------
1040  T = T + 1
---------------------
1050  GOTO 1000
---------------------
1999  REM EVENT ROUTINE
---------------------
2000  IF WY = 10 THEN 2030
---------------------
2009  REM ADD WEIGHT
---------------------
2010  FOR I = 1 TO INC
    : HPLOT WX,WY
    : WX = WX - 2
    : IF WX = 83 THEN WY = WY - 2
  * : WX = 103
  * : IF WY = 9 THEN 2030
---------------------
2020  NEXT
---------------------
2030  IF Y > 55 THEN 2500
---------------------
2199  REM DAMAGE THRESHOLD MOVEMENT
---------------------
2200  VTAB 7
    : HTAB X / 7 + 1
    : PRINT "7"
```

```
     : HCOLOR= 1
     : HPLOT 13,48 TO 266,48
     : HCOLOR= 5
     : FOR I = 13 TO 266 STEP 5
     : HPLOT I,55
     : NEXT
     : HCOLOR= 3
     : FOR I = 1 TO 12
     : J = PEEK (SS)
     : NEXT
----------------------
2210  VTAB 2
     : HTAB 13
     : FOR I = 1 TO 3
     : PRINT SEESAW$(I);
     : FOR J = 1 TO 500
     : NEXT
     : PRINT BLNK$;
     : NEXT
     : PRINT SEESAW$(4);
     : FOR I = 1 TO 3000
     : NEXT
     : PRINT BLNK$;SEESAW$(0);
----------------------
2499  REM SET UP FOR NEXT EVENT
----------------------
2500  T = - 15 * LOG ( RND (1))
----------------------
2510  T1 = INT (T) + 1
----------------------
2520  T = - 10 * LOG ( RND (1))
----------------------
2530  INC = INT (T) + 1
----------------------
2540  T = 0
----------------------
2550  IF Y > 55 THEN RETURN
----------------------
2560  VTAB 7
     : HTAB X / 7 + 1
     : PRINT L$ + "Z" + K$
     : HCOLOR= 1
     : HPLOT 13,48 TO 266,48
     : HCOLOR= 5
     : FOR I = 13 TO 266 STEP 5
     : HPLOT I,55
     : NEXT
     : HCOLOR= 3
     : VTAB 8
     : HTAB 3
     : PRINT V$;
```

```
    : VTAB 19
    : HTAB 36
    : PRINT W$;
    : PRINT P$;Y$
    : WX = 103
    : WY = 29
    : RETURN
---------------------
2999  REM KEYPRESS SUBROUTINE
---------------------
3000  IF PEEK ( - 16384) = 209 THEN POKE - 16368,0
  * : PRINT Y$;P$;Z$
  * : END
---------------------
3010  IF PEEK ( - 16384) = 205 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN MENU"
---------------------
3020  IF PEEK ( - 16384) = 200 THEN PRINT Z$
  * : PRINT CHR$ (4);"RUN DAMAGE HELP"
---------------------
3030  CS = PEEK (115) + PEEK (116) * 256
---------------------
3040  IF PEEK ( - 16384) = 177 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD POISSON.SET,A",CS
  * : PRINT CHR$ (4);"RUN POISSON PROCESS"
---------------------
3050  IF PEEK ( - 16384) = 178 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD QUEUE.SET,A",CS
  * : PRINT CHR$ (4);"RUN MM1Q"
---------------------
3060  IF PEEK ( - 16384) = 179 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD DAM.SET,A",CS
  * : PRINT CHR$ (4);"RUN DAM MODEL"
---------------------
3070  IF PEEK ( - 16384) = 181 THEN PRINT Z$
  * : PRINT CHR$ (4);"BLOAD FISH.SET,A",CS
  * : PRINT CHR$ (4);"RUN FISHPOND"
---------------------
3080  IF PEEK ( - 16384) < > 160 THEN 3110
---------------------
3090  IF PEEK ( - 16384) = 160 THEN POKE - 16368,0
  * : REM CLOCK STOPPED TO PAUSE
---------------------
3100  IF PEEK ( - 16384) < > 160 THEN 3100
  * : REM AWAITING SPC BAR TO RESUME
---------------------
3110  POKE - 16368,0
    : RETURN
---------------------
9999  REM INITIALIZE
---------------------
```

```
10000 A$ = CHR$ (1)
    : B$ = CHR$ (2)
    : C$ = CHR$ (3)
    : D$ = CHR$ (4)
    : K$ = CHR$ (11)
    : L$ = CHR$ (12)
    : P$ = CHR$ (16)
    : Q$ = CHR$ (17)
    : V$ = CHR$ (22)
    : W$ = CHR$ (23)
    : Y$ = CHR$ (25)
    : Z$ = CHR$ (26)
----------
10010 DIM SEESAW$(4)
----------
10020 SEESAW$(0) = B$ + "   " + C$ + "   " + C$ + "890
          ABA" + C$ + " JKLMNOP     ABA" + C$ + "       3Q
        RSTUVWX" + Q$ + D$
----------
10030 SEESAW$(1) = B$ + "456" + C$ + "GHG" + C$ + "DEF
          ABA" + C$ + " JKLMNOP     ABA" + C$ + "       3Q
        RSTUVWX" + Q$ + D$
----------
10040 SEESAW$(2) = B$ + L$ + "STS" + C$ + K$ + "GHG
        " + L$ + "UVU" + C$ + K$ + "GHG          ABA" + C$
        + "2YZ" + L$ + "ABCWD     NOP" + C$ + "       YGXIJ
        KLM" + K$ + Q$ + D$
----------
10050 SEESAW$(3) = B$ + "             " + L$ + "STS" + C$ + "
        UVU          " + K$ + "GHG" + C$ + "ABA          GH
        G" + C$ + L$ + "NOP     DWHBQR" + K$ + "Y1" + C$ +
        + L$ + " MLEJFXGY" + K$ + Q$ + D$
----------
10060 SEESAW$(4) = B$ + "             456" + C$ + "
          GHG" + C$ + "ABA          DEF" + C$ + "ABA     P
        ONMLKJ" + C$ + "IWCUTSRQ3" + Q$ + D$
----------
10070 BLNK$ = B$ + "                " + C$ + "
         " + C$ + "                 " + C$ + "
          " + C$ + "                  " + Q$ + D$
----------
10080 POKE - 16368,0
    : PRINT Z$;O$;A$;O$;D$;P$;A$;"1"
----------
10090 HCOLOR= 1
    : HPLOT 13,152 TO 267,152 TO 267,48 TO 13,48 TO 13,152
    : FOR I = 55 TO 150 STEP 10
    : HPLOT 11,I TO 13,I
    : NEXT
    : FOR I = 23 TO 266 STEP 20
    : HPLOT I,152 TO I,154
```

```
     : NEXT
     : HCOLOR= 5
     : FOR I = 13 TO 266 STEP 5
     : HPLOT I,55
     : NEXT
     : HCOLOR= 3
     : VTAB 2
     : HTAB 13
     : PRINT SEESAW$(0)
----------------------
10100 SS = - 16336
     : WX = 103
     : WY = 29
     : X = 14
     : Y = 151
----------------------
10110 GOSUB 2500
     : REM INITIALIZE FIRST EVENT
----------------------
10120 RETURN
```

E. FISHPOND LOGO PROGRAM

```
---------------------
10    GOSUB 10000
    : REM INITIALIZE
---------------------
999   REM MAIN ROUTINE; CLOCK
---------------------
1000  IF TT = T1 THEN GOSUB 2000
  * : GOTO 1000
---------------------
1010  FOR I = 1 TO R
    : NEXT
---------------------
1020  IF PEEK ( - 16384) > 127 THEN GOSUB 3000
  * : REM KEY PRESSED TO STOP CLOCK
---------------------
1030  TT = TT + 1
    : GOTO 1000
---------------------
1999  REM FISH JUMP SUBROUTINE
---------------------
2000  FOR I = 1 TO HT
    : HTAB X
    : VTAB Y
    : PRINT UP$
    : Y = Y - 1
    : NEXT
---------------------
2010  HTAB X
    : VTAB Y
    : PRINT BLNK$
    : HTAB X
    : VTAB Y
    : PRINT STAY$
    : HTAB X
    : VTAB Y
    : PRINT BLNK$
---------------------
2020  FOR I = 1 TO HT
    : HTAB X
    : VTAB Y
    : PRINT DOWN$
    : Y = Y + 1
    : NEXT
---------------------
2030  HTAB X
    : VTAB Y
    : PRINT BLNK$
    : HTAB X
    : VTAB Y
```

```
    : PRINT SPLSH$
    : HTAB X
    : VTAB Y
    : PRINT BLNK$
---------------------
2040  X = INT (30 * RND (1) + 4)
---------------------
2049  REM SET UP FOR NEXT JUMP
---------------------
2050  Y = INT (3 * RND (1) + 20)
---------------------
2060  HT = INT ( - 10 * LOG ( RND (1)))
    : IF HT > 5 THEN HT = 5
---------------------
2070  T = - LOG (1 - RND (1))
---------------------
2080  T1 = INT (T) + 1
---------------------
2090  TT = 0
---------------------
2100  RETURN
---------------------
2999  REM KEYPRESS SUBROUTINE
---------------------
3000  IF PEEK ( - 16384) = 211 THEN R = 1000
  * : GOTO 3070
  * : REM RUN CLOCK SLOW
---------------------
3010  IF PEEK ( - 16384) = 198 THEN R = 1
  * : GOTO 3070
  * : REM RUN CLOCK FAST
---------------------
3020  IF PEEK ( - 16384) = 160 THEN 4000
---------------------
3030  IF PEEK ( - 16384) = 193 THEN 4200
---------------------
3040  IF PEEK ( - 16384) = 201 THEN 4400
---------------------
3050  IF PEEK ( - 16384) = 205 THEN PRINT Z$;
  * : VTAB 7
  * : HTAB 1
  * : PRINT V$;
  * : VTAB 19
  * : HTAB 40
  * : PRINT W$;O$;A$;P$;O$;B$;P$;O$;T$;P$;Z$
  * : PRINT CHR$ (4);"RUN MENU"
---------------------
3060  IF PEEK ( - 16384) = 209 THEN POKE - 16368,0
  * : PRINT Z$;P$
  * : END
---------------------
```

```
3070  POKE - 16368,0
    : RETURN
----------------------
3999  REM PROGRAM MENU SUBROUTINE
----------------------
4000  PRINT Z$
    : VTAB 7
    : HTAB 1
    : PRINT V$;
    : VTAB 19
    : HTAB 40
    : PRINT W$;O$;A$;P$;O$;B$;P$;O$;T$;P$
----------------------
4010  VTAB 8
    : HTAB 35
    : PRINT A$"2";S$;"KEY";C$;C$; TAB( 3);S$;"ACKNOWLEDGEMEN
       TS";K$;".................A"
----------------------
4020  PRINT TAB( 3);S$;"INTRODUCTION";K$;"...................
       ..I";C$; TAB( 3);S$;"MENU";K$;"....................
       .........M"
----------------------
4030  PRINT Y$;O$;A$;P$;A$"1"
    : GOTO 3070
----------------------
4099  REM ACKNOWLEDGEMENTS SUBROUTINE
----------------------
4200  PRINT Z$;P$
    : VTAB 3
    : HTAB 9
    : PRINT "-[ ";S$;"ACKNOWLEDGEMENTS ]-"
----------------------
4210  VTAB 21
    : HTAB 11
    : PRINT "[SPC] TO CONTINUE"
----------------------
4220  VTAB 4
    : HTAB 1
    : PRINT V$;
    : VTAB 20
    : HTAB 40
    : PRINT W$;
----------------------
4230  VTAB 5
    : HTAB 3
    : PRINT S$;"THE AUTHOR WISHES TO ACKNOWLEDGE";C$; TAB( 3
       );"THE USE OF THE FOLLOWING SOFTWARE";C$; TAB( 3);"
       IN PRODUCING THIS PACKAGE;"
----------------------
4240  PRINT C$;C$; TAB( 3);K$;"'APPLESOFT TOOL KIT', ";L$;"BY
        ";S$;"APPLE ";C$; TAB( 3);S$;"COMPUTER, ";S$;"INC.
```

```
"
---------------------
4250  PRINT C$; TAB( 3);K$;"'PRINTER II', ";L$;"BY ";S$;"COMP
        UTER ";S$;"SYSTEMS ";C$; TAB( 3);S$;"DESIGN"
---------------------
4260  POKE - 16368,0
---------------------
4270  IF PEEK ( - 16384) < > 160 THEN 4270
---------------------
4280  PRINT P$
    : VTAB 5
    : HTAB 3
    : PRINT S$;"THE ";S$;"APPLESOFT TOOL KIT PROVIDED";C$;
        TAB( 3);"THE MEANS FOR BUILDING, DISPLAYING,";C$;
        TAB( 3);"AND ANIMATING CHARACTERS, PLUS A";C$; TAB(
        3);"HOST OF OTHER FEATURES TO ENHANCE"
---------------------
4290  PRINT TAB( 3);"PROGRAM DISPLAY.";C$;C$; TAB( 3);S$;"THE
         ";S$;"PRINTER ";K$;"II ";L$;"PACKAGE PROVIDED";C$;
        TAB( 3);"FORMATTING OF NUMERICAL OUTPUT FOUND";C$;
        TAB( 3);"IN TWO PROGRAMS OF THIS PACKAGE."
---------------------
4300  POKE - 16368,0
---------------------
4310  IF PEEK ( - 16384) < > 160 THEN 4310
---------------------
4320  PRINT Y$;P$
    : GOTO 4000
---------------------
4399  REM INTRODUCTION SUBROUTINE
---------------------
4400  PRINT Z$;P$
    : VTAB 3
    : HTAB 11
    : PRINT "-[ ";S$;"INTRODUCTION ]-"
---------------------
4410  VTAB 21
    : HTAB 11
    : PRINT "[SPC] TO CONTINUE"
---------------------
4420  VTAB 4
    : HTAB 1
    : PRINT V$;
    : VTAB 20
    : HTAB 40
    : PRINT W$;
---------------------
4430  VTAB 5
    : HTAB 3
    : PRINT S$;"THE FOUR PROGRAMS OF THIS PACKAGE";C$; TAB(
        3);"SIMULATE SITUATIONS WHICH ARE BASED";C$; TAB( 3
```

```
);"ON AN UNDERLYING ";S$;"POISSON PROCESS."
---------------------
4440  PRINT C$; TAB( 3);S$;"THE FIRST, LABELED '";S$;"POISSON
        ";S$;"PROCESS',";C$; TAB( 3);"IS A SIMPLE DEMONSTR
      ATION OF THE";C$; TAB( 3);"OCCURRENCE OF EVENTS IN
      A ";S$;"POISSON";C$; TAB( 3);"MANNER.  ";S$;"THE SE
      COND, '";K$;"MM1Q',"
---------------------
4450  PRINT TAB( 3);L$;"SIMULATES A SIMPLE SINGLE-";C$; TAB(
      3);"SERVER QUEUE WITH EXPONENTIAL";C$; TAB( 3);"INT
      ERARRIVAL TIMES AND EXPONENTIAL";C$; TAB( 3);"SERVI
      CE TIMES, (THE";K$;" M/M/1 ";L$;"QUEUE).";
---------------------
4460  POKE - 16368,0
---------------------
4470  IF PEEK ( - 16384) < > 160 THEN 4470
---------------------
4480  PRINT P$
    : VTAB 5
    : HTAB 3
    : PRINT S$;"THE THIRD, THE '";S$;"DAM ";S$;"MODEL', PORT
      RAYS";C$; TAB( 3);"ONE OF THE CLASSIC APPLICATIONS
      OF";C$; TAB( 3);"THE COMPOUND ";S$;"POISSON PROCESS
      .  ";S$;"THE"
---------------------
4490  PRINT TAB( 3);"FOURTH PROGRAM, THE '";S$;"DAMAGE ";S$;"
      MODEL',";C$; TAB( 3);"DEPICTS THE COMPOUND ";S$;"PO
      ISSON PROCESS";C$; TAB( 3);"APPLIED TO THE CUMULATI
      VE DAMAGE";C$; TAB( 3);"MODEL."
---------------------
4500  PRINT
    : PRINT TAB( 3);S$;"THE PROGRAMS ARE DESIGNED TO PROVIDE
      ";C$; TAB( 3);"A REAL TIME PERCEPTION OF ";S$;"POIS
      SON ";C$; TAB( 3);"PROCESSES IN ACTION."
---------------------
4510  POKE - 16368,0
---------------------
4520  IF PEEK ( - 16384) < > 160 THEN 4520
---------------------
4530  PRINT P$
    : VTAB 5
    : HTAB 3
    : PRINT S$;"THE MENU PROVIDES THE DETAILS OF";C$; TAB( 3
      );"HOW TO RUN EACH OF THE PROGRAMS.";C$; TAB( 3);S$
      ;"ANY PROGRAM MAY BE RUN FROM ANY"
---------------------
4540  PRINT TAB( 3);"OTHER PROGRAM BY PRESSING THE KEY";C$;
      TAB( 3);"DESIGNATED IN THE MENU."
---------------------
4550  PRINT C$; TAB( 3);S$;"FINALLY, EACH PROGRAM IS PROVIDED
      ";C$; TAB( 3);"WITH ITS OWN '";S$;"HELP' PROGRAM WH
```

```
ICH";C$; TAB( 3);"DEFINES KEYBOARD COMMANDS.  ";S$;
        "PRESS";C$; TAB( 3);"THE ";
---------------------
4560  PRINT "'";S$;"H' KEY TO GET '";S$;"HELP.'"
---------------------
4570  POKE - 16368,0
---------------------
4580  IF PEEK ( - 16384) < > 160 THEN 4580
---------------------
4590  PRINT Y$;P$
    : GOTO 4000
---------------------
4699  REM TITLE SUBROUTINE
---------------------
4700  PRINT O$;T$;P$;O$;B$;A$"1"
    : VTAB 4
    : HTAB 3
    : PRINT ID$;C$;C$;C$; TAB( 11);L$;A$"2";"BY  ";K$;"R. J.
        ";S$;"DAVISON";C$;C$;
---------------------
4710  PRINT C$; TAB( 3);S$;"SUBMITTED IN PARTIAL FULFILLMENT"
        ;C$; TAB( 3);"OF THE REQUIREMENTS FOR THE DEGREE";C
        $; TAB( 3);"OF ";S$;"MASTER OF ";S$;"SCIENCE IN ";S
        $;"OPERATIONS";C$; TAB( 3);S$;"RESEARCH FROM THE ";
---------------------
4720  PRINT S$;"NAVAL ";S$;"POSTGRADUATE";C$; TAB( 3);S$;"SCH
        OOL, ";S$;"MONTEREY, ";S$;"CA.   ";S$;"ADVISORS; ";
        C$; TAB( 3);K$;"J. D. ";S$;"ESARY AND ";K$;"J. L. "
        ;S$;"ELLIS."
---------------------
4730  PRINT C$;C$; TAB( 11);"[SPC] TO CONTINUE";O$;A$;P$;K$;A
        $;"1"
    : RETURN
---------------------
9999  REM INITIALIZE SUBROUTINE
---------------------
10000 A$ = CHR$ (1)
    : B$ = CHR$ (2)
    : C$ = CHR$ (3)
    : D$ = CHR$ (4)
    : E$ = CHR$ (5)
    : K$ = CHR$ (11)
    : L$ = CHR$ (12)
    : O$ = CHR$ (15)
    : P$ = CHR$ (16)
    : Q$ = CHR$ (17)
    : S$ = CHR$ (19)
    : T$ = CHR$ (20)
    : V$ = CHR$ (22)
    : W$ = CHR$ (23)
    : Y$ = CHR$ (25)
```

```
    : Z$ = CHR$ (26)
----------------------
10010 UP$ = B$ + "AB" + C$ + "HI" + C$ + "  " + Q$ + D$
----------------------
10020 STAY$ = B$ + "CD" + C$ + "JK" + Q$ + D$
----------------------
10030 DOWN$ = B$ + "  " + C$ + "EF" + C$ + "LM" + Q$ + D$
----------------------
10040 SPLSH$ = B$ + "   " + C$ + " NO" + C$ + "PQR" + Q$ + D$
----------------------
10050 BLNK$ = B$ + "   " + C$ + "   " + C$ + "   " + Q$ + D$
----------------------
10060 ID$ = B$ + " 1 2 4 3 3 2 5   1 6 2 7 % 3 3 % 3" + C$ +
        " 8 9 ! 0 0 9 (   8 # 9 $ & 0 0 & 0" + Q$ + D$
----------------------
10070 POKE - 16368,0
----------------------
10080 R = 1000
    : SPEED= 235
----------------------
10090 GOSUB 4700
    : REM DISPLAY TITLE
----------------------
10100 GOSUB 2040
    : REM INITIALIZE 1ST JUMP
----------------------
10110 RETURN
```

F. MENU PROGRAM

```
----------------------
1000  A$ = CHR$ (1)
    : C$ = CHR$ (3)
    : D$ = CHR$ (4)
    : K$ = CHR$ (11)
    : L$ = CHR$ (12)
    : P$ = CHR$ (16)
    : S$ = CHR$ (19)
    : V$ = CHR$ (22)
    : W$ = CHR$ (23)
    : Y$ = CHR$ (25)
    : Z$ = CHR$ (26)
----------------------
1010  PRINT Z$;P$;A$"2"
    : POKE - 16368,0
    : HCOLOR= 3
    : HPLOT 1,24 TO 279,24
    : HPLOT 1,136 TO 279,136
    : VTAB 6
    : HTAB 5
    : PRINT V$;
    : VTAB 16
    : HTAB 30
    : PRINT W$;
----------------------
2000  VTAB 6
    : HTAB 11
    : PRINT "-[ ";K$;"MENU ]-"
    : VTAB 7
    : HTAB 26
    : PRINT S$;"PRESS"
----------------------
2010  VTAB 8
    : HTAB 1
    : PRINT S$;"POISSON ";S$;"PROCESS";K$;".............1";C$
        ;"MM1Q.....................2";C$;S$;"DAM ";S$;"MO
        DEL.................3"
----------------------
2020  VTAB 11
    : HTAB 1
    : PRINT S$;"DAMAGE ";S$;"MODEL..............4";C$;S$;"F
        ISHPOND ";S$;"LOGO.............5";C$;S$;"QUIT";K$;
        ".....................Q"
----------------------
2030  VTAB 15
    : HTAB 16
    : PRINT S$;"SELECTION -[ ]-";
----------------------
2040  VTAB 15
```

```
    : HTAB 28
    : GET R$
    : VTAB 15
    : HTAB 28
    : PRINT VAL (R$);
    : FOR I = 1 TO 1000
    : NEXT
--------------------
2050  IF VAL (R$) = 0 THEN POKE - 15368,0
  * : PRINT Y$;P$;Z$
  * : END
--------------------
2060  I = VAL (R$)
    : CS = PEEK (115) + PEEK (116) * 256
    :
    : ON I GOTO 3000,3010,3020,3030,3040
--------------------
2070  GOTO 2040
--------------------
3000  PRINT Z$
    : PRINT CHR$ (4);"BLOAD POISSON.SET,A",CS
    : PRINT CHR$ (4);"RUN POISSON PROCESS"
--------------------
3010  PRINT Z$
    : PRINT CHR$ (4);"BLOAD QUEUE.SET,A",CS
    : PRINT CHR$ (4);"RUN MM1Q"
--------------------
3020  PRINT Z$
    : PRINT CHR$ (4);"BLOAD DAM.SET,A",CS
    : PRINT CHR$ (4);"RUN DAM MODEL"
--------------------
3030  PRINT Z$
    : PRINT CHR$ (4);"BLOAD DAMAGE.SET,A",CS
    : PRINT CHR$ (4);"RUN DAMAGE MODEL"
--------------------
3040  PRINT Z$
    : PRINT CHR$ (4);"BLOAD FISH.SET,A",CS
    : PRINT CHR$ (4);"RUN FISHPOND"
```

G. SAMPLE HELP PROGRAM

```
---------------------
1000  A$ = CHR$ (1)
    : C$ = CHR$ (3)
    : D$ = CHR$ (4)
    : K$ = CHR$ (11)
    : L$ = CHR$ (12)
    : P$ = CHR$ (16)
    : S$ = CHR$ (19)
    : V$ = CHR$ (22)
    : W$ = CHR$ (23)
    : Y$ = CHR$ (25)
    : Z$ = CHR$ (26)
---------------------
1010  PRINT Z$;P$;A$"2"
    : POKE - 16368,0
    : HCOLOR= 3
    : HPLOT 1,24 TO 279,24
    : HPLOT 1,136 TO 279,136
    : VTAB 6
    : HTAB 5
    : PRINT V$;
    : VTAB 16
    : HTAB 32
    : PRINT W$;
---------------------
2000  VTAB 6
    : HTAB 9
    : PRINT "-[ ";K$;"HELP ]-"
---------------------
2010  VTAB 8
    : HTAB 1
    : PRINT S$;"THIS PROGRAM PRESENTS TWO INDE- PENDENT ";S$
       ;"POISSON PROCESSES.      ";S$;"INITIALLY THEY ARE
       COMPOSED AS  ONE.  ";S$;"THEY MAY BE DECOMPOSED AND
       RUN TOGETHER OR RUN SEPARATELY."
---------------------
2020  VTAB 15
    : HTAB 1
    : PRINT S$;"     [SPC] TO CONTINUE";
---------------------
2030  POKE - 16368,0
---------------------
2040  IF PEEK ( - 16384) < > 160 THEN 2040
---------------------
2050  PRINT P$
    : VTAB 6
    : HTAB 3
    : PRINT "-[ ";K$;"KEYBOARD COMMANDS ]-"
    : VTAB 7
```

```
    : HTAB 25
    : PRINT S$;"KEY"
----------
2060  VTAB 8
    : HTAB 1
    : PRINT S$;"CLOCK ";S$;"FAST";K$;".............F";C$;S
       $;"CLOCK ";S$;"SLOW (DEFAULT)";K$;".....S";C$;S$;"M
       ENU";K$;".....................M";C$;S$;"QUIT";K$;".
       .....................Q"
----------
2070  VTAB 12
    : HTAB 1
    : PRINT S$;"CHANGE ";S$;"MEANS";K$;".............C";C$;S
       $;"PAUSE/";S$;"RESUME.........SPC BAR";C$;C$;"
        [SPC] TO CONTINUE";
----------
2080  POKE - 16368,0
----------
2090  IF PEEK ( - 16384) < > 160 THEN 2090
----------
2100  PRINT P$
    : VTAB 6
    : HTAB 3
    : PRINT "-[ ";K$;"KEYBOARD COMMANDS ]-"
    : VTAB 7
    : HTAB 25
    : PRINT S$;"KEY"
----------
2110  VTAB 8
    : HTAB 1
    : PRINT S$;"EVENTS ";S$;"SUPERIMPOSED";K$;"......U";C$;S
       $;"EVENTS ";S$;"DECOMPOSED";K$;".......T";C$;S$;"A
       UTO ";S$;"EVENTS ";S$;"ONLY";K$;".........A";C$;S$;
       "BUS ";S$;"EVENTS ";S$;"ONLY";K$;".........B"
----------
2120  VTAB 15
    : HTAB 1
    : PRINT S$;"     [SPC] TO CONTINUE";
----------
2130  POKE - 16368,0
----------
2140  IF PEEK ( - 16384) < > 160 THEN 2140
----------
2150  PRINT P$
    : VTAB 6
    : HTAB 3
    : PRINT "-[ ";K$;"KEYBOARD COMMANDS ]-"
    : VTAB 7
    : HTAB 26
    : PRINT S$;"KEY"
----------
```

```
2160  VTAB 8
    : HTAB 1
    : PRINT S$;"STATISTICS ";S$;"DISPLAY";K$;".......D";C$;
        S$;"STATISTICS ";S$;"DISPLAY ";S$;"OFF";K$;"....O";
        C$;S$;"VEHICLE ";S$;"DISPLAY ";S$;"OFF";K$;".......
        N";C$;S$;"TIME ";S$;"DISPLAY ";S$;"OFF";K$;".......
        ...N"
----------------------
2170  VTAB 12
    : HTAB 1
    : PRINT S$;"RESUME ";S$;"VEHICLE ";S$;"DISPLAY";K$;"....
        Y";C$;S$;"RESUME "S$;"TIME ";S$;"DISPLAY ";K$;"....
        ..Y";C$;C$;L$;"      [SPC] TO CONTINUE";
----------------------
2180  POKE - 16368,0
----------------------
2190  IF PEEK ( - 16384) < > 160 THEN 2190
----------------------
2200  PRINT P$
    : VTAB 6
    : HTAB 3
    : PRINT "-[ ";K$;"KEYBOARD COMMANDS ]-"
----------------------
2210  VTAB 8
    : HTAB 1
    : PRINT S$;"THE STATISTICS DISPLAY MUST";C$;"BE ON AND V
        EHICLE DISPLAY OFF";C$;"BEFORE THE TIMING DISPLAY";
        C$;"CAN BE TURNED OFF.  ";S$;"REVERSE";C$;"THE ORDE
        R TO RESUME NORMAL";C$;"OPERATIONS."
----------------------
2220  POKE - 16368,0
----------------------
2230  VTAB 15
    : HTAB 1
    : PRINT "[SPC] TO REVIEW, ";K$;"[Q] ";L$;"TO QUIT";
----------------------
2240  IF PEEK ( - 16384) = 160 THEN PRINT P$
  * : POKE - 16368,0
  * : GOTO 2000
----------------------
2250  IF PEEK ( - 16384) = 209 THEN POKE - 16368,0
  * : PRINT Z$
  * : PRINT CHR$ (4);"RUN POISSON PROCESS"
----------------------
2260  GOTO 2240
```

H. STARTUP PROGRAM

```
----------
99    REM PROGRAM TO LOAD HRCG
----------
100   HOME
    : HGR
----------
110   ONERR GOTO 370
----------
120   ADRS = 0
----------
130   PRINT CHR$ (4);"BLOAD RBOOT"
----------
140   CALL 520
    : REM EXECUTE RBOOT
----------
150   ADRS = USR (0),"HRCG"
----------
199   REM BRING IN HRCG, ADRES=STARTING ADDRESS
----------
200   IF ADRS < = 0 THEN ADRS = ADRS + 65536
  * : REM MAKE ADRS POSITIVE
----------
210   CS = ADRS - 768 * 2
    : HIMEM: CS
----------
220   CH = INT (CS / 256)
    : CL = CS - 256 * CH
----------
230   POKE ADRS + 7,CL
    : POKE ADRS + 8,CH
----------
240   PRINT CHR$ (4);"BLOAD FISH.SET,A";CS
----------
250   PRINT CHR$ (4);"BLOAD ROMAN.SET,A";CS + 768
----------
300   CALL ADRS
    : REM INITIALIZE HRCG
----------
310   PRINT CHR$ (4);"BLOAD P2,A$9A07"
    : REM  LOAD PRINTER II
----------
320   CALL 39431
    : REM  INITIALIZE PRINTER II
----------
330   POKE 216,0
    : REM ONERR OFF
----------
340   PRINT CHR$ (4);"BLOAD FISHPOND.PIC,A$4000"
----------
```

```
350   PRINT CHR$ (4);"RUN FISHPOND"
----------------------
360   END
----------------------
370   PRINT "UNABLE TO LOAD"
----------------------
380   POKE 216,0
    : REM ONERR OFF
----------------------
390   END
```

BIBLIOGRAPHY

Apple Computer, Inc., APPLE II Reference Manual, 1979

Apple Computer, Inc., Applesoft Tool Kit Manual, 1980

Apple Computer, Inc., Basic Programming Reference Manual, 1978

Apple Computer, Inc., The DOS Manual, 1980

Cinlar, Erhan, Introduction to Stochastic Processes, Prentice-Hall, Inc., 1975

Haight, Frank A., Handbook of the Poisson Distribution, John Wiley and Sons, Inc., 1967

Lewis, P. A. W., and Shedler, G. S., "Simulation of Non-Homogeneous Poisson Processes with Log-Linear Rate Function," Biometrika, v. 63, 1976

Lewis, P. A. W. and Shedler, G. S., "Simulation of Nonhomogeneous Poisson Processes by Thinning," Naval Research Logistics Quarterly, v. 26, no. 3, September 1979

Moran, P. A. P., The Theory of Storage, Wiley and Sons, Inc., 1959

Naval Postgraduate School Report 55EY73041A, Multivariate Geometric Distributions Generated by a Cumulative Damage Process, by J. D. Esary and A. W. Marshall, March 1973

Naval Postgraduate School Report 55EY73071A, Families of Components, and Systems, Exposed to a Compound Poisson Damage Process, by J. D. Esary and A. W. Marshall, July 1973

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Professor J. D. Esary, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5.	Maj. Jeffery L. Ellis, USA Office of the Deputy Chief of Staff for Operations and Plans (WOZ2AA) Pentagon, Washington, DC 20310	1
6.	Commandant(G-PTE) U.S. Coast Guard Washington, DC 20590	2
7.	LCDR Michael E. Finley, USN USS Nimitz (CVN-68) FPO New York 09542	1
8.	Capt. Thomas Pawlowski, USA Department of Mathematical Sciences (DFMS) U.S. Air Force Academy, Colorado 80840	1
9.	LCDR Richard J. Davison, USCG Commandant (G-DMT-1) U.S. Coast Guard Washington, DC 20590	2